

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



FACULTE DE GENIE DE LA CONSTRUCTION

DEPARTEMENT DE GENIE MECANIQUE

Mémoire de Fin d'Etude

MASTER

Domaine : Science Et Technologie

Filière : Electromécanique

Spécialité : Maintenance Industrielle

Présenté par
Sam ibtissem

encadré par
Mr SI AHMED.H

Thème

**Conception et réalisation d'un smart robot
pour la
Détection des flammes et des objets**

Mémoire soutenu publiquement le devant le jury composé de :

M r. BELGAID .H

M C B, UMMTO, Président

M r. YAMANI .

M C B, UMMTO, Examineur

M r. SI AHMED.H

M A A, UMMTO, Encadreur

2020/2021

Remerciements

*Avant tout, je remercie Dieu qui m'a donné la force et le courage
pour accomplir ce travail.*

*Je remercie, et très chaleureusement mon promoteur, monsieur
H. SIAHMED pour son aide et ses conseils pour la réalisation de
ce mémoire.*

*Je remercie également les membres de jury qui feront honneur de
juger mon travail et d'apporter leurs réflexions.*

*Et tous les enseignants qui ont contribué et participé à ma
formation.*

*Mes plus chaleureux remerciements pour ceux qui ont apporté
leur soutien et leur aide de près ou de loin pour réussir ce
travail.*

Ibtissem SAM.

Dédicaces

Je dédie ce modeste travail:

A mes parents que dieu les préserve en bonne santé et une longue vie. Qui n'ont jamais cessé d'être à mes cotés et m'encourager pour réussir dans mes études.

A ma grand -mère.

A ma sœur et mes frère.

A toute ma famille.

A tous mes amis (es) sans exceptions.

Ainsi que mes collègues de ma promotion master maintenance industrielle.

Merci à tous

Ibtissem SAM.

Sommaire

Introduction générale :	1
Cahier des charges :	3
Chapitre 1 : les robots mobiles	
I. INTRODUCTION.....	4
I.1.Définition.....	4
I.1.1 .Le robote	4
I.1.2.La robotique.....	4
I.2.L’historique de la robotique.....	4
I.3.Les types de robots.....	5
I.3.1. Robotes manipulateurs	5
I.3.2. Robots explorateurs.....	5
I.4.Classification des robots mobiles	5
I.4.1. Classification selon le degré d’autonomie.....	5
I.4.1.a. Robot télécommandé	5
I.4.1.b. Robot semi-autonome	5
I.4.1.c.Robot autonome	5
I.4.2. Classification selon le domaine d’application	5
I.4.3. Classification selon le type de locomotion	6
I.4.3.a.Les robots mobiles à roues.....	6
I.4.3.b.Les robots à chenilles.....	8
I.4.3.c.Les robots marcheurs.....	9
I.5.Avantage et inconvénient de quelques types de robots	9
I.6.Conclusion	10

Chapitre 2 : conception du système

II. Introduction.....	11
II.1 .présentation de l' Arduino	11
II.1.1. définition de l' Arduino.....	11
II.1.2. domaine d'application de l' Arduino	12
II.1.3. Bonnes raisons de choisir Arduino.....	12
II.1.4.Les types de cartes Arduino.....	13
II.2.Présentation d' Arduino Uno	13
II.2.1.caractéristiques de l' Arduino Uno.....	14
II.2.2.Les sources de l'alimentation de la carte	14
II.2.3.Les entrées & sorties	15
II.2.4.Le Microcontrôleur ATmega328	16
II.3.Les capteurs utilisés.....	17
II.3.1.Capteur de lumière LDR (photorésistance)	17
II.3.1.a. description du capteur LDR.....	17
II.3.1.b.Principe de fonctionnement	17
II.3.1.c. les caractéristiques de la photorésistance LDR.....	18
II.3.2. Capteur à ultrasons HC-SR04.....	18
II.3.2.a.description du capteur à ultrasons HC-SR04.....	18
II.3.2.b.Principe de fonctionnement du Capteur à Ultrasons HC-SR04.....	19
II.3.2.c.Les caractéristiques du Capteur à Ultrasons HC-SR04.....	19
II.3.3.capteur de température et d'humidité.....	19
II.3.3.a.définition de DHT22.....	19
II.3.3.b.les caractéristiques de capteur DHT22.....	20
II.3.3.c.le câblage de DHT22.....	20
II.4. le Buzzer.....	21
II.5. Shields DK electronics.....	22

II.5.1.Caractéristiques de Shield DK électronique.....	22
II.6.Moteur à courant continu (MCC).....	22
II.7.Servomoteurs.....	23
II.7.1.Fonctionnement du servomoteur.....	24
II.7.2.Caractéristiques.....	24
II.8.Le module Bluetooth HC-05.....	25
II.8.1.Pourquoi choisir le module Bluetooth HC-05.....	25
II.9.L'environnement de programmation Arduino.....	25
II.9.1Fenêtre générale de l'application Arduino.....	26
II.10.Logiciel ISIS PROTEUS	27
II.10.1. Définition	27
II.10.2.fonctionnement de ISIS PROTEUS.....	28
II.10.3.Avantages de l'utilisation de ISIS PROTEUS.....	28
II.11.La camera ESP32-cam.....	28
II.11.1.définition d'ESP32-CAM.....	28
II.11.2.fonctionnement d' ESP32-CAM	29
II.11.3.les composants de l'ESP32-cam.....	30
II.11.4.les caractéristiques de ESP32.....	31
II.11.5.installation et programmation.....	32
II.13. Le Châssis	35
II.13.1.Caractéristiques.....	35
II.14. Conclusion	36

Chapitre 3 : Asservissement d'un robot mobil

III. Introduction.....	37
III.1.les systèmes de vision dans la robotique	37
III.1.1.les système de vision.....	37

III.1.1.a.la vision monoculaire	37
III. 1.1.b.la stéréovision.....	38
III.1.1.c.la vision omnidirectionnelle.....	38
III.2.Modélisation de la caméra	39
III.2.1.conception	39
III.2.2.composants et la modalisation d'une caméra.....	40
III.3 L'asservissement visuel.....	40
III.3.1.l'algorithme de poursuite d'une trajectoire.....	40
III.3.1.a. L'acquisition et traitement d'image.....	40
III.3.1.b. extraction de la caractéristique.....	40
III.3.1.c. algorithme de traitement d'image	41
III.3.2.Résultats de simulation de traitement d'image.....	47
III.3.3.choix des primitives visuelles.....	47
III.4. conclusion.....	48

Chapitre 4 : Amélioration du robot

IV. Introduction.....	49
IV.1.L'application Androïde.....	49
IV.1.1.présentation Android.....	49
IV.1.2.présentation de MIT APP Inventor.....	49
IV.2.Matériel.....	51
IV.3.Création de l'application AI2.....	51
IV.3.1.Les étapes de création de l'Application.....	51
IV.4.Programmation de l'Application AI2.....	56
IV.5.Compiler et partager votre Application.....	58
IV. 6.Amélioration de la précision de robot grâce au capteur ultrason.....	62
IV. 6.1.Mesure de la distance par les ULTRASON.....	62

IV.6.2. Méthode d'application	63
IV.6.3. Calcule de la distance.....	64
IV.6.4. <i>Application sur Arduino</i>	64
IV.6.5. Programmation.....	65
IV.6.6. Schématisation sur ISIS PROTEUS.....	65
IV.7. Amplification de l'alimentation du Robot.....	66
IV.7.1. Convertisseur Boost	66
IV.7. 2.Pacifications	67
IV.7.3. Exemple de code	67
IV.3.4. Exemple de câblage.....	68
IV.CONCLUSION.....	69
Conclusion générale.....	70
Annexe	
Bibliographie	

Listes des figures :

Figure I.1 :les robots manipulateurs.....	3
Figure I.2 :pack bote utilise a Fukushima.....	4
Figure I.3 : Robot agricole.....	4
Figure I.4 : robot unicycle	7
Figure I.5 : robot tricycle	7
Figure I.6 : robot omnidirectionnel	7
Figure I.7 : robot voiture quatre roues	8
Figure I.8 : robot à chenille	8
Figure I.9 : robot marcheur	9
Figure II.10 : Système Arduino	11
Figure II.11 : Arduino Uno	13
Figure II.12 : Microcontrôleur ATmega328	16
Figure II.13 : la photorésistance.....	17
Figure II.14 : fonctionnement de la LDR.....	18
Figure II.15 : Capteur à Ultrasons HC-SR04.....	18
Figure II.16 : Principe de fonctionnement du Capteur à Ultrasons HC-SR04.....	19
Figure II.17 :capteur DHT 22.....	20
Figure II.18 :cablage d'un DHT 22.....	21
Figure II.19 : le Buzzer.....	21
Figure II.20 : Shield DK electronics(L298N).....	22
Figure II.21: Moteur à courant continu	23
Figure II.22 : servomoteur.....	24
Figure II.23 : module Bluetooth HC-05.....	24
Figure II.24 : application Arduino 1.8.1.....	25
Figure II.25 : Interface d'utilisateur du logiciel proteus	27

Figure II.26 : schéma de fonctionnement de ISIS Proteus	27
Figure II.27 : système ESP32-cam	28
Figure II.28 : principe de fonctionnement	29
Figure II.29 :les composons d'ESP32-cam.....	29 30
Figure II.30 :la fiche technique d'ESP32-CAM.....	31
Figure II.31 : le châssis	35
Figure III.32 : champ de vue des systèmes de vision.....	37
Figure III.33 : exemple de capteur catadioptrique.....	39
Figure III.34 : formation d'une image dans un sténopé.....	40
Figure III.35 : algorithme de traitement d'image.....	41
Figure III.36 : l'image original.....	47
Figure III.37 :l'image en niveaux de grix.....	47
Figure III.38 : le plan rouge de l'image	47
Figure III.39 : identification de la trajectoire.....	47
Figure III.40 :l'image après filtrage et binarisation.....	47
Figure III.41 :la squelettisation de trajectoire	47
Figure IV.42 : logo Android.....	50
Figure IV.43 : MIT APP INVENTOR.....	51
Figure IV.44 : site de MIT APP INVENTOR.....	52
Figure IV.45 : fenêtre pour nome le projet.....	53
Figure IV.46 : fenêtre des projets.....	54
Figure IV.47 : interface de la création APP Inventor.....	54
Figure IV.48 : une fenêtre qui nous présente ces composons.....	55
Figure IV.49 : Aligner des objets.....	56
Figure IV. 50 : BluetoothClien.....	56
Figure IV.51 : un HorizontalArrangement avec des boutons	57

Liste des tableaux :

Tableau [I.1] : domaine d'application des robots.	6
Tableau [I.2] : Avantages et inconvénients des robots mobiles.....	10
Tableau [II.3]: caractéristique d'Arduino Uno.....	
Tableau [II.4]: Caractéristiques techniques du capteur HC-SR04.....	20
Tableau [II.5]: Barre d'actions IDE Arduino.....	26
Tableau [III.6] : Les caractéristiques du châssis.....	32

Liste des abréviations :

RUR : Rossum's Universel Robots (Robots Universel de Rossum).

I.N.S.A : Institut National des sciences appliquées.

APP : Application.

MIT : Massachusetts Institute of Technology (Institut de technologie Massachusetts).

IDE : integrated development environment (environnement de développement libre).

PWM : Pulse Width Modulation (Modulation de largeur d'impulsion).

USB: Universal Serial Bus.

ICSP: In Circuit Serial Programming.

VIN : La tension d'entrée.

SPI: Interface Série Périphérique.

LED: light emitting diode.

RAM: Random Access Memory (memoire vive).

EEPROM: Electrically Erasable Programmable Read-Only Memory.

GND: Ground.

E / S : Entrées/Sorties.

LPG: Liquified Petroleum Gas (GPL).

RCO : Rapport Cyclique d'Ouverture.

Wifi : Wireless Fidelity (fidélité sans fil).

VCC : Alimentation Tension Continue.

DC : Direct Current : courant continu.

Tx-Rx : Transmission et Réception.

LDR : light-dépendent résistor (résistance photo-dépendante).

DHT 22 : Digital-output relative Humidity and Temperature sensor.

MCC: moteur a courant continue.

ESP32-cam : est un module de camera.

APP : application.

App Inventor : C'est un logiciel en ligne qui permet de créer des applications pour appareils Android

AI2 : application inventore.

ESP-WROOM-32 : est un puissant module MCU générique Wi-Fi+Bluetooth+BLE qui s'adresse à une grande variété d'applications, allant des réseaux de capteurs à faible puissance aux tâches les plus exigeantes, telles que le codage de la voix, la diffusion de musique en continu et le décodage de MP3

AI thinker : le nom de fabriquons de l'ESP-32-CAM.

GPIO: sont des ports d'entrées/sorties très utilisés dans le monde des microcontrôleurs en particulier dans le domaine de l'électronique embarquée, elles sont placées sur un circuit électronique afin de communiquer avec des composants électroniques et circuits externes. Il peut s'agir de détecteurs ou senseurs pour capter des données, ou encore de contrôler des commandes

UART : *Universal Asynchronous Receiver Transmitter*, est un émetteur-récepteur asynchrone universel

Fablab : laboratoire de fabrication

Sténopé : est un dispositif optique très simple permettant d'obtenir un appareil photographique dérivé de la chambre noire. Il s'agit d'un trou de très faible diamètre percé dans une plaque de très faible épaisseur. Par extension, on appelle ainsi l'appareil photographique utilisant un tel dispositif

Cloud : désigne les serveurs accessibles sur Internet, ainsi que les logiciels et bases de données qui fonctionnent sur ces serveurs. Le cloud permet aux utilisateurs d'accéder aux mêmes fichiers et aux mêmes applications à partir de presque n'importe quel appareil

Application cloud : Une application cloud, ou « appli cloud », est un programme logiciel dans lequel fonctionnent ensemble des composants locaux ou installés en cloud.

Scratch : est un langage de programmation graphique manipulable et exécutable par le logiciel de même nom à vocation éducative permet de modifier le code du programme en cours d'exécution.

Code QR: un type de code-barres à deux dimensions (ou code matriciel) format optique lisible par machine pouvant être visualisé sur l'écran d'un appareil mobile ou imprimé sur papier, constitué de modules-carrés noirs disposés dans un carré à fond blanc.

APK : signifie « **Android Package Kit** » : cela représente un format de fichiers conçu pour le système d'exploitation Android. Le fichier APK contient tous les éléments nécessaires à l'installation d'une application sur votre Smartphone Android.

AIA : est un document qui contient la facturation du contrat terminé tout au long du projet; ce document a été créé principalement pour aider les architectes travaillant sur un projet de construction à être payés pour les travaux réalisés sur le projet.

Avant -propos

La science se construit brique par brique .chacun y apporte le sienne, ce mémoire est une continuation du le travail de M. Hamani Djamel.

Sont thème de fin d'étude en Master spécialité : Maintenance Industrielle qui a obtenu en 2020 a L'UUMTO.

Intitulé : conception et réalisation d'un smart robot a base de la carte Arduino [ref].

Pour cela je conseille aux étudiants de lire le mémoire de M. Hamani Djamel avant d'entamer le mien.

Résumé :

L'objectif de ce projet est de concevoir et de réaliser une commande à distance pour un robot mobile à chenille, à l'aide d'une carte Arduino Uno .Le but est que le robot puisse se déplacer dans un environnement statique et d'indiquer la température et de pouvoir visualiser les terrains pour l'observateur, cette machine est dotée de fonctions de perception, de décision et d'action afin d'être capable d'effectuer des tâches diverses et d'accomplir correctement sa tâche. En outre nous proposons une application Android pour pouvoir choisir le mode de contrôle du robot.

Mots clés :

Robot mobile – Arduino Uno – système Android - Robot autonome.

Abstract:

The objective of this project is to design and realize a remote control for a caterpillar mobile robot, using an Arduino Uno board. The objective is that the robot can move in a static environment and indicate the temperature and to be able to visualize the grounds for the observer, this machine is equipped with functions of perception, decision and action in order to be able to carry out various tasks and to accomplish its task correctly. In addition, we offer an Android application to be able to choose the robot's control mode.

Keywords : Mobile robot - Arduino Uno - Android system - Autonomous robot.

Introduction générale

INTRODUCTION GENERALE

La robotique est une branche qui comprend l'électronique, la mécanique et l'informatique. Aujourd'hui, le monde connaît une évolution importante dans ces domaines, ce qui a permis de concevoir et de réaliser des robots aptes à être utilisés dans divers buts, tel que réaliser des tâches complexes, pénibles, répétées ou dans des endroits qui sont inaccessibles ou dangereux pour l'homme. Citons par exemple : la détection et la désactivation des bombes.

Avec l'aide d'un système commandé, on peut apporter des solutions à des problèmes sans risques de perdre des vies humaines ou d'avoir des problèmes de santé. Pour cela, il faut mettre tout le nécessaire dans le système comme la camera, qui va permettre de visualiser la zone d'intervention en temps réel et placer différents capteurs qui vont relever tous les détails sur l'environnement, par ailleurs en utilisant des moteurs de précision comme le servomoteur on obtient des résultats meilleurs.

Ainsi, le contexte dans lequel s'inscrit notre travail, consiste à concevoir et réaliser un robot mobile avec trois modes de contrôle, qui répond au cahier des charges. Dans l'objectif de donner au robot la capacité qui va lui permettre de se mouvoir dans son environnement, on a utilisé un châssis à chenilles. Avec deux applications Android on peut choisir le mode de contrôle de notre châssis sans fil en utilisant une transmission Bluetooth et Wifi (camira).

Pour pouvoir déplacer le robot, on a utilisé trois modes différents :

- **Mode 1** : le robot navigue en autonomie, avec la possibilité d'éviter les obstacles.
- **Mode 2** : contrôler le robot en fonction d'une intensité lumineuse.
- **Mode 3** : mode caméra avec la détection de la température

Une camera est fixée sur le châssis du robot, celle-ci permet à l'utilisateur de visualiser le site de navigation + le contrôle.

Le présent mémoire est organisé comme suit :

- Le premier chapitre, contient des généralités sur la robotique, une présentation générale sur les robots mobiles.
- Le second chapitre est consacré à l'étude de la commande du robot mobile, un aperçu général sur le matériel utilisé, du logiciel et l'application androïde.
- Le troisième chapitre est consacré à l'asservissement visuel d'un robot mobile.
- Le quatrième chapitre est consacré à la création de l'application MIT APP INVENTOR + l'amélioration de la position du Robot et la détection de la température + humidité.

Et on termine avec une conclusion générale.

Cahier des charges

Cahier des charges :

Nous avons fixé notre propre cahier des charges en proposant ce sujet. L'idée était donc de créer notre propre robot mobile idéalement commandé à distance. Nous avons donc pu distinguer les différentes fonctions de notre cahier des charges comme suit :

Les fonctions de ce robot sont :

Fonction principale : permettre à l'homme d'intervenir sur une zone difficilement accessible.

Fonction 1 : navigation du robot en autonomie avec possibilité d'éviter les obstacles.

Fonction 2 : contrôler le robot en fonction d'une intensité lumineuse.

Fonction 3 : acquérir une image en temps réel sur l'environnement et une commande normal avec un amplificateur.

Fonction 4 : disposer d'une autonomie d'énergie suffisante avec un amplificateur

Fonction 5 : crée une application MIT APP INVANTOR

❖ Le matériel utilisé :

- Une carte ArduinoUno.
- Une carte de puissance (L298N) pour contrôler les moteurs du robot.
- Deux moteurs à courant continu.
- Module Bluetooth (HC-05) pour la communication entre le robot et le Smartphone.
- Une camera pour transmettre l'image(ESP32-CAM) + commande normal.
- Un servomoteur.
- Un capteur ultrason.
- Deux photorésistances (capteurs de lumière).
- Une batterie pour alimenter le robot.
- Un capteur de température et d'humidité (DHT22).
- Un ordinateur.
- Un Smartphone.
- Boosteur (amplificateur du courant).

Chapitre I

Généralité sur les robots

I. INTRODUCTION

Le terme « robot » a été introduit en 1920 par l'écrivain tchèque Karel Èapek. Ce terme, provenant du tchèque robota, « travail forcé », désigne à l'origine une machine androïde capable de remplacer l'homme dans toutes ses tâches. ultérieurement, pendant les années 70 et 80, un nouveau domaine émergea dans l'automatisme : la robotique. Ce dernier a connu un développement considérable profitant des avancées technologiques des autres domaines tels que la microélectronique, les microprocesseurs et les capteurs. [1] Depuis une vingtaine d'années, un effort particulier a été fait dans les domaines de la recherche et de l'industrie pour construire des robots mobiles évoluant avec un minimum d'intervention humaine. Une première génération des robots a consisté en des machines capables d'évoluer dans des environnements parfaitement connus : celles-ci réalisent des missions planifiées à partir d'une modélisation complète de l'environnement (laboratoire) ou se contentent de suivre une trajectoire par un mécanisme de filo-guidage (robot de manutention). Le point commun de ces robots est qu'ils évoluent dans un environnement qui leur est totalement dédié. [2]

La robotique a un rôle important dans la vie de l'être humain, elle est une activité multidisciplinaire visant l'étude, la conception et la construction de robots ou plus simplement de machines automatiques. Sa pratique réunit des savoir-faire techniques et des connaissances scientifiques des domaines de l'électronique, de l'informatique et de la mécanique. La robotique est utilisée dans de multiples domaines : la robotique industrielle, domestique, scientifique, éducative, médicale, militaire, loisir et le transport...etc. Elle se subdivise en deux types : les robots industriels et les robots mobiles. Les robots industriels sont généralement fixes, ils sont utilisés dans de nombreuses applications industrielles: l'assemblage mécanique, la soudure, la peinture... Tandis que les robots mobiles ne sont pas fixes, ils sont classifiés selon la locomotion en robots marcheurs, à roues, à chenilles...etc. Comme ils peuvent être classifiés selon le domaine d'application en robots militaires, de laboratoire industriel et de services.

I.1.Définition

I.1.1. Le robot

- Le robot est un dispositif mécanique articulé capable d'imiter certaines fonctions humaines telles que la manipulation d'objets ou la locomotion, dans le but de se substituer à l'homme pour la réalisation de certaines tâches matérielles, cette réalisation est plus ou moins autonome selon les facultés de perception de l'environnement dont est doté le robot.
- D'après le Petit Larousse, un robot est un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable. En fait, l'image que chacun se fait d'un robot est généralement vague. [1]
- Un robot est un appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe ou modifiable.

I.1.2.La robotique

C'est l'ensemble des activités de construction et de mise en œuvre des robots, on peut dire aussi que tout dispositif comporte une partie opérationnelle qui réalise la tâche et une partie décisionnelle ou commande qui contrôle la partie opérationnelle. Comme il peut être défini comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles. [3]

I.2.L'histoire de la robotique

La robotique a connue plusieurs générations :

- 1947 : Premier manipulateur électrique télé-opéré.
- 1954 : Premier robot programmable.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot.
- 1981 : Un robot équipé d'un arceau de sécurité pour la détection d'obstacles dans un environnement inconnu. Nommé le robot VESA, construit à l'I.N.S.A (France).
- 2005 : création d'un robot à quatre pattes nommé BIG DOG.

I.3. Types de robots [4]

On a deux grandes catégories de robots sont :

I.3.1 Robots manipulateurs : robots ancrées physiquement à leur place de travail et généralement mis en place pour réaliser une tâche précise ou répétitive. Dans cette catégorie on trouve plusieurs types de robots sont

➤ **Des robots de complexité variable**

Automates : le robot exécute une même série d'actions indéfiniment, sans aucune perception de son environnement,

Robots réactifs : le robot exécute une action selon l'état actuel de l'environnement dans lequel il évolue (le mapping état/action étant fixé initialement),

Robots cognitifs : le robot analyse son environnement et calcule la meilleure action à effectuer.

Robots industriels : chaînes de montage, test qualité, manipulation de produits chimiques,...

Robots pour l'assistance médicale : est un système robotique utilisé dans le cadre d'une application thérapeutique.[4]



Figure I.1:les robots manipulateurs

I.3.2 Les robots mobiles

Les robots mobiles ont une place particulière en robotique. Leur intérêt réside dans leur mobilité qui ouvre des applications dans de nombreux domaines. Ce robot doté de moyens de locomotion qui lui permettent de se déplacer suivant son degré d'autonomie, il peut être doté de moyens de perception et de raisonnement. [3]

➤ **Robots explorateurs**

Ils sont destinés à explorer des environnements où l'homme ne peut pas se rendre.

- exploration d'autres planètes comme Mars : Séjourner, Spirit.

- exploration d'épaves ou de décombres : recherche de victimes aux World Trade Centuyyhuer ou lors de tremblements de terre.
- déminage de terrains.
- exploration de zones radioactives : entretien de réacteurs, de piscine de stockage, etc...



Figure I.2: pack bot utilise a Fukushima

➤ Robots de service

Ils sont destinés à aider l'homme pour certaines tâches.

- robots agricoles.
- robots de transport de marchandises : robots docker.
- robots ménagers : aspirateur, tondeuse.
- guide de musée: cité de l'espace à Toulouse.
- aide aux personnes : personnes âgées ou ayant un handicap.



Figure I.3 : robot agricole

➤ **Robots ludiques**

- Aibo, Qrio, ...
- Mindstorm .

➤ **Robots humanoïdes**

- Asimo.
- HRP.
- Nao.

I.4. Classification des robots mobiles :

La classification des robots mobiles se fait suivant plusieurs critères :

I.4.1. Classification selon le degré d'autonomie :

Un robot mobile autonome est un système automoteur doté de capacités décisionnelles et de moyens d'acquisition et de traitement de l'information qui lui permettent d'accomplir sous contrôle humain réduit un certain nombre de tâches dans un environnement non complètement connu c'est classification la plus intéressante. [4]

On peut citer quelques types:

I.4.1.a. Robot télécommandé

Ce sont des robots commandés par un opérateur (machine ou l'homme), qui leurs dicte chaque tâche élémentaire à faire (avancer, reculer, tourner à droite, etc.).

I.4.1.b. Robot semi-autonome

Le robot peut effectuer des taches prédéfinis sans l'aide de l'operateur.

I.4.1.c. Robot autonome

C'est un robot qui réalise des tâches semi-définies. Ce type de robot pose des problèmes d'un niveau de complexité élevé de représentation des connaissances, de capacité décisionnelle et de génération de plans qui sont résolus à bord dans la mesure du possible.

I.4.2. Selon le domaine d'application

Les travaux en robotique ont pour but de concevoir et de construire des machines capables d'évoluer et d'interagir avec un environnement physique de manière à accomplir les différentes tâches pour lesquelles elles ont été créés. à titre d'exemple :

Domains	Applications
Industries nuclear	- surveillance de sites - manipulation de matériaux radioactive - dismantlement de centrals
Secrete civil	- neutralization d'activité terrorist - déminage - pose d'explosif - surveillance de munitions
Chimique	- surveillance de sites - manipulation de matériaux toxiques
Mine	- assistance d'urgence
Agricole	- cueillette de fruits - traite, moisson, traitement des vignes.
Nettoyage	- coque de navire - nettoyage industriel
Espace	- exploration
Militaire	-surveillance - pose d'explosif - manipulation de munition

Tableau [I.1] : domaine d'application des robots. [3]

I.4.3. Selon le type de locomotion

I.4.3.a. Les robots mobiles à roues

La mobilité par roues est la structure mécanique la plus utilisée. Ce type de robot assure un déplacement aisé, mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues, déterminées principalement, par la position et le nombre de roues utilisées. Nous citerons ici les quatre classes principales de robots à roues. [3]

➤ **Robots unicycle**

Un robot de type unicycle est actionné soit par une seule roue ou deux roues indépendantes. On utilise des capteurs d'attitude et de déplacement pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

Un robot unicycle est un robot qui bouge dans un plan 2D ayant une certaine vitesse allant d'avant, mais zéro mouvement latéral instantané, car les robots unicycle sont des systèmes non-holonome. Il est impossible de le déplacer dans une direction perpendiculaire aux roues de

locomotion. [3]

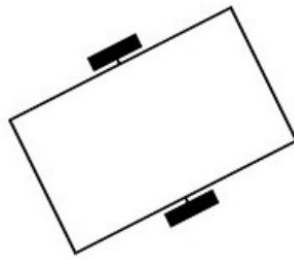


Figure I.4 : robot unicycle.

➤ Robots tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. [3]

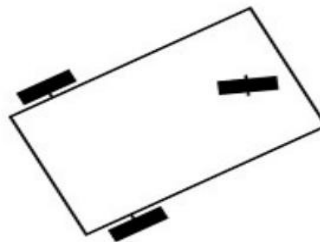


Figure I.5 : robot tricycle.

➤ Robots mobiles omnidirectionnels

Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses : vitesse de translation selon les axes x et y et vitesse de rotation autour de z . Il peut se déplacer librement dans toutes les directions. Il est généralement constitué de trois roues décentrées orientables

L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande. [3]

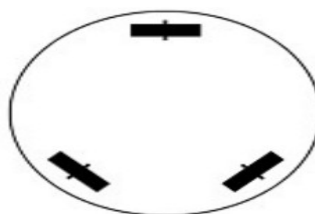


Figure I.6 : robot omnidirectionnel.

➤ Robot voiture quatre roues

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe. Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé. [3]

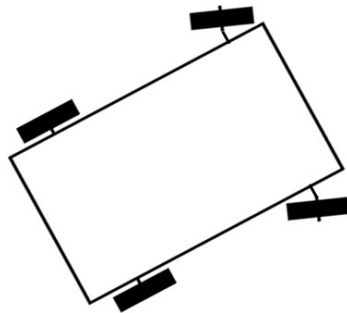


Figure I.7 : Robot voiture quatre roues.

I.4.2.b. Les robots à chenilles

L'utilisation des chenilles présente l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence. [3]

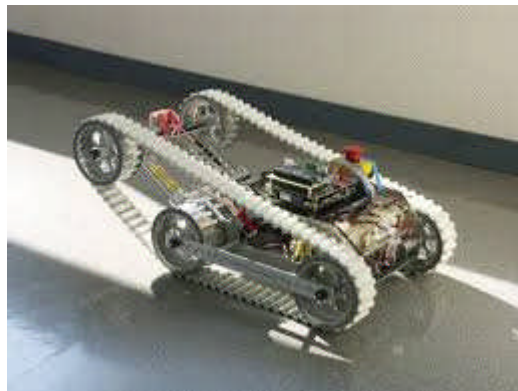


Figure I.8 : Robot à chenille.

I.4.2.c. Les robots marcheurs

Les robots marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile. Leur anatomie à nombreux degrés de liberté permet un rapprochement avec les robots manipulateurs. La locomotion est commandée en termes de coordonnées articulaires. Les différentes techniques étudiées se rapprochent de la marche des animaux et notamment de celle

des insectes.



Figure I.9 : Robot marcheur.

I.5. Avantage et inconvénient de quelques types de robots :

Type de robot	Avantages	Inconvénients
Robot unicycle	<ul style="list-style-type: none"> -stable -rotation sur soi-même -complexité mécanique faible 	<ul style="list-style-type: none"> -non holonome
Robot tricycle	<ul style="list-style-type: none"> -complexité mécanique modérée 	<ul style="list-style-type: none"> -non holonome -peu stable -pas de rotation sur soi-même
Robot voiture	<ul style="list-style-type: none"> -complexité mécanique modérée -stable 	<ul style="list-style-type: none"> -non holonome
Robot omnidirectionnel	<ul style="list-style-type: none"> -holonome -stable -Rotation sur soi-même 	<ul style="list-style-type: none"> -complexité mécanique importante
Robot à chenille	<ul style="list-style-type: none"> -stable -franchit les obstacles 	<ul style="list-style-type: none"> -usure rapide -vitesse faible -bruit élevé

Tableau [I.2] : Avantages et inconvénients des robots mobiles. [5]

I.6.Conclusion :

Le robot peut être manipulé à plusieurs degrés de liberté, à commande automatique et reprogrammable. Dans ce premier chapitre, nous avons présenté le robot de façon générale, et on a bien détaillé les robots mobiles.

Chapitre 2: Matériel utilise pour la conception du Robot

II. Introduction

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore, aujourd'hui, l'électronique est de plus en plus remplacée par l'électronique programmée. Pour programmer des systèmes électroniques, et parmi ces systèmes on utilise souvent le système Arduino afin de simplifier les schémas et optimiser le coût de réalisation.

Dans ce chapitre nous allons voir une présentation générale du système esp32-cam et de Arduino UNO, ses caractéristiques et une présentation de l'application Android MIT APP INVENTOR et on mettra la lumière sur les logiciels IDE Arduino et ISIS PROTEUS.

II.1.Présentation de l'Arduino

II.1.1.Définition de l'Arduino

Arduino est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le pilotage d'un robot, la domotique (le contrôle des appareils domestiques éclairage, chauffage..etc). Elle se présente généralement sous la forme d'une carte électronique bleue, qui à peu près la taille d'une main. Cette carte comporte des inscriptions en blanc qui permettent de repérer ses différents éléments. Tous les composants et les circuits de la carte sont visibles et accessibles.[9]

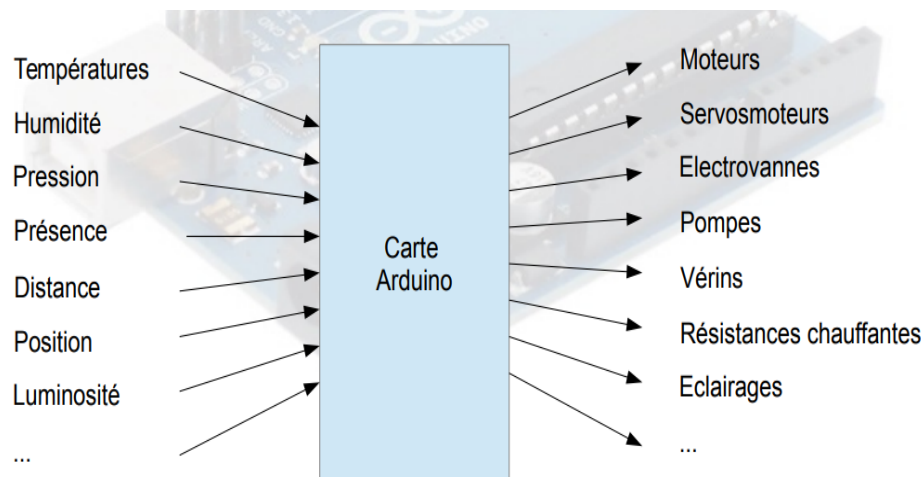


Figure II.10 : carte Arduino.

II.1.2. Domaine d'application de l'Arduino

L'utilisation de la carte arduino est diversifiée, il permet de réaliser beaucoup de choses dans plusieurs domaines. On peut citer quelques exemples :

- Fabriquer des robots.
- Contrôler des appareils domestiques.
- Télécommander des appareils mobiles.
- Utiliser le jeu de lumière.
- Communiquer avec l'ordinateur.

II.1.3. Bonnes raisons de choisir Arduino

Pour la programmation, il existe plusieurs plateformes basées sur des microcontrôleurs qui permettent de faire la même chose. Nous allons voir pourquoi choisir l'Arduino.

a. Le prix

Les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent moins de 2500 DA).

b. Open source

L'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également. Le logiciel et le langage sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.

c. La compatibilité

Le logiciel, tout comme la carte, sont compatibles sous systèmes d'exploitation courants (Windows, Linux et Mac), contrairement aux autres outils de programmation du commerce qui ne sont, en général, compatibles qu'avec Windows.

d. La communauté

La communauté d'arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet. De plus, on trouve les références du langage Arduino ainsi qu'une page complète de tutoriels sur le site arduino.cc (en anglais).

II.1.4. Les types de cartes Arduino

Il y a trois types de cartes :

- ❖ Les dites « officielles » qui sont fabriquées en Italie par le fabricant officiel : Smart Projects.
- ❖ Les dits « compatibles » qui ne sont pas fabriqués par Smart Projects, mais qui sont totalement compatibles avec les Arduino officielles.
- ❖ Les « autres » fabriquées par diverse entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino ...).

II.2. Présentation d'Arduino Uno

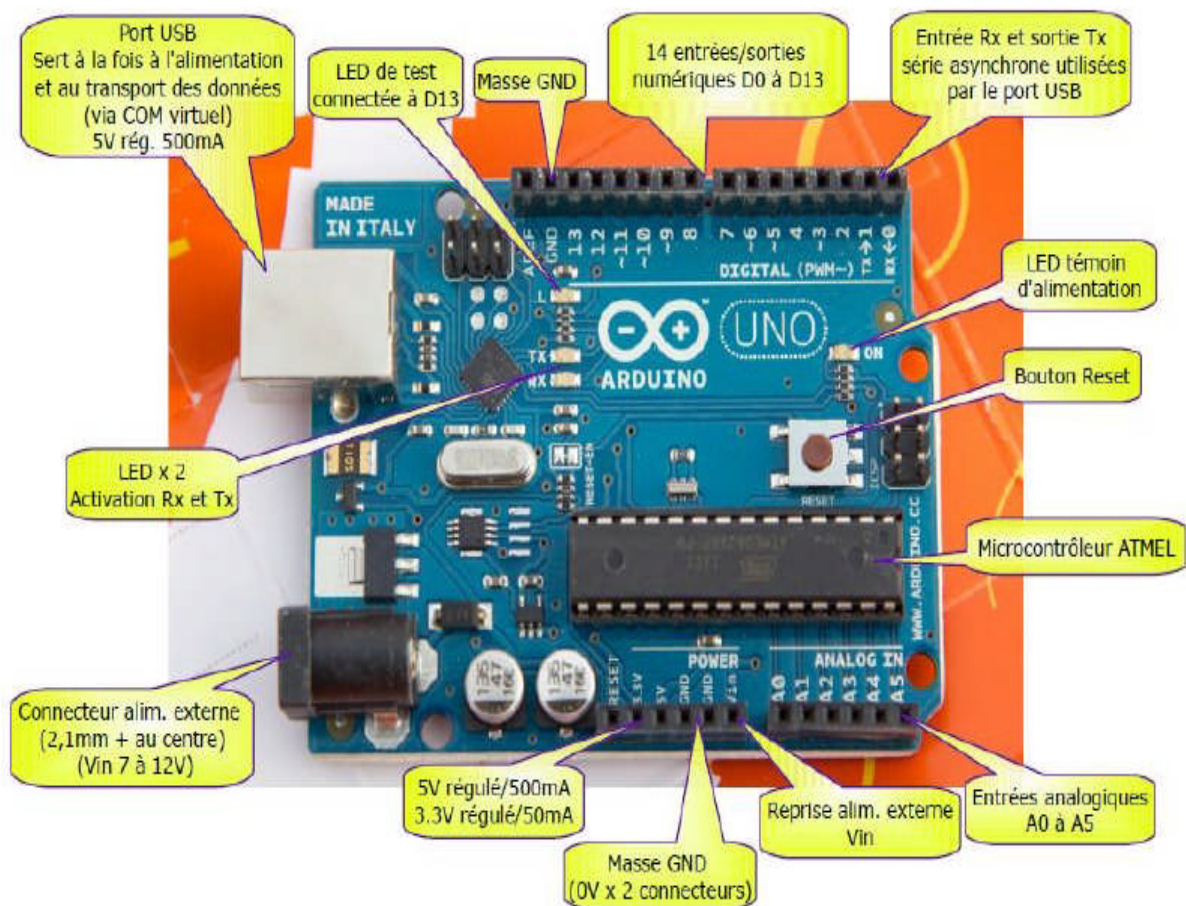


Figure II.11: Arduino Uno

II.2.1. caractéristiques de l'Arduino Uno

Microcontrôleur	ATmega328 (8 bits)
Tension de fonctionnement	5 V
Tension d'alimentation (recommandée)	7- 12 V
Tension d'alimentation (limites)	6 - 20V
Nombre d'E/S	14 (dont 6 pouvant générer des signaux PWM)
Nb ports "Analogique/Numérique"	6
Courant max. par E/S	40 mA sous +5V
Courant pour broches 3.3 V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Flash	32 Ko (ATmega328) dont 0.5 Ko utilisé par le bootloader
SRAM	2 Ko (ATmega328)
EEPROM	1 Ko (ATmega328)
Vitesse horloge	16 MHz

Tableau [II.3]: caractéristiques d'Arduino Uno

II.2.2. Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée /Sortie) et cela comme suit :

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée).
- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (les circuits électroniques numériques nécessitent)

une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- **3,3V** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [11]

II.2.3. Les entrées & sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 Kohms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes**: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI** (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13(SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- **I2C**: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C(ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`).
- **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches

mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le Court circuit ou la surcharge soit stoppé.[12]

II.2.4.Le Microcontrôleur ATmega328

Un microcontrôleur se compose des transistors des résistances et des condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la (figure II.10) montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.



Le composant classique



Le composant CMS

Figure II.12: Microcontrôleur ATmega328

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de :

- **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 22 ko.

- **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme

II.3. Les capteurs utilisés

II.3.1. Capteur de lumière LDR (photorésistance) [13]

Une photorésistance est un composant électrique dont la résistivité varie en fonction de la quantité de la lumière incidente. On peut également la nommer résistance photo-dépendante (light-dépendent resistor (LDR) ou cellule photoconductrice.



Figure II.13 : la photorésistance

II.3.1.1. Principe de fonctionnement

Une photorésistance est composée d'un semi-conducteur à haute résistivité. Si la lumière incidente est de fréquence suffisamment élevée (donc d'une longueur d'onde inférieure à la longueur d'onde seuil), elle transporte une énergie importante. Au-delà d'un certain niveau propre au matériau, les photons absorbés par le semi-conducteur donneront aux électrons liés assez d'énergie pour passer de la bande de valence à la bande de conduction. La compréhension de ce phénomène entre dans le cadre de la théorie des bandes. Les électrons libres et les trous d'électron ainsi produits abaissent la résistance du matériau.

Lorsque le photon incident est suffisamment énergétique, la production des paires électron-trou est d'autant plus importante que le flux lumineux est intense. La résistance évolue donc comme l'inverse de l'éclairement, cette relation peut être considérée comme linéaire sur une plage d'utilisation limitée.

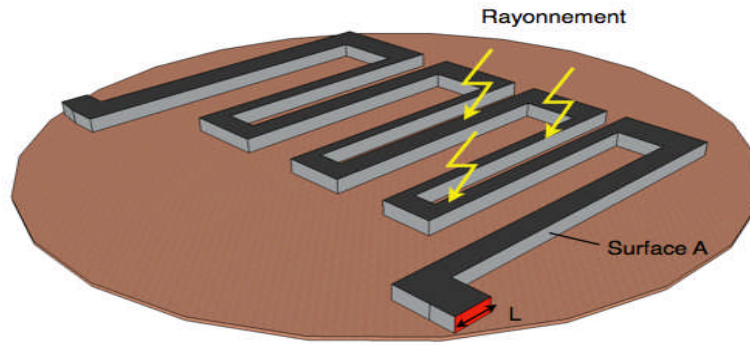


Figure II.14 : fonctionnement de la LDR

II.3.1.2.les caractéristiques de la photorésistance LDR

- Faible coût.
- Larges gammes spectrales.
- Facilité de mise en œuvre.
- Rapport de transfert statique.
- Sensibilité élevée.
- Temps de réponse de (0,1 us à 100 ms).

II.3.2. Capteur à ultrasons HC-SR04

II.3.2.1.description du capteur à ultrasons HC-SR04

Le HC-SR04 est un capteur de distance utilise les ultrasons, Il offre une excellente plage de détection avec des mesures stables. Son fonctionnement, n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.

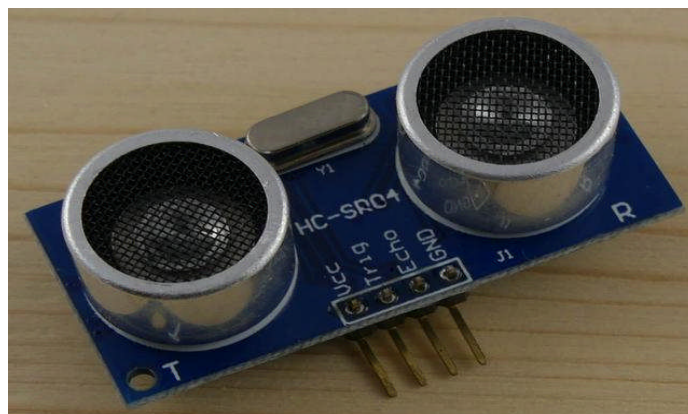


Figure II.15 : Capteur à Ultrasons HC-SR04.

II.3.2.2.Principe de fonctionnement du Capteur à Ultrasons HC-SR04

Le capteur émet de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur.

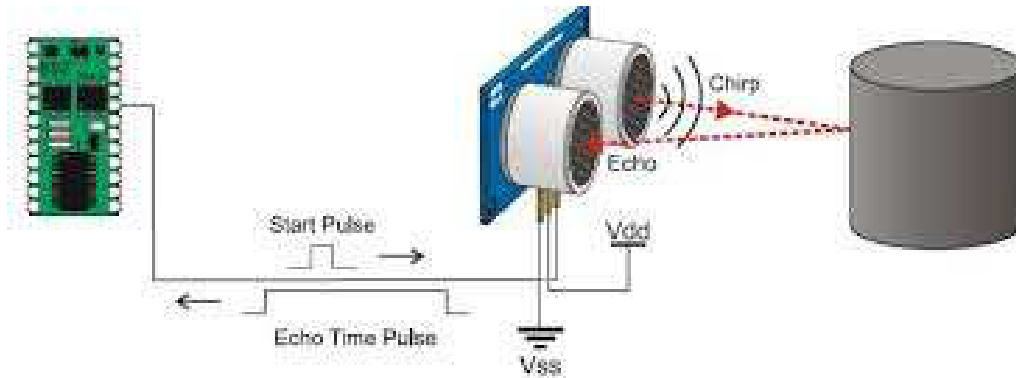


Figure II.16 : Principe de fonctionnement du Capteur à Ultrasons HC-SR04

II.3.2.3.Les caractéristiques du Capteur à Ultrasons HC-SR04

Caractéristique	Valeur
Dimensions	45 mm x 20 mm
Plage de mesure	2 cm à 4m
Résolution de la mesure	0.3 cm
Angle de mesure efficace	<15°
Largeur d'impulsion	10 μs
Fréquence	40KHZ

Tableau [II.3]: Caractéristiques techniques du capteur HC-SR04.

II.3.3. Capteurs de température et d'humidité DHT22

II.3.3.1. définition de DHT22

DHT c'est pour "Digital-output relative Humidity & Temperature sensor", ce qui signifie capteur d'humidité et température à sortie numérique.[6]

Le DHT22 est un capteur numérique de base . A faible coût permettant de mesurer de manière efficace la température et l'humidité de l'air ambiant grâce à sa combinaison deux en un d'un capteur d'humidité capacitif et d'une thermistance.

Le capteur de température et d'humidité DHT22 (ou AM2302) communique avec un microcontrôleur via un port série. Le capteur est calibré et ne nécessite pas de composants supplémentaires pour pouvoir être utilisé.

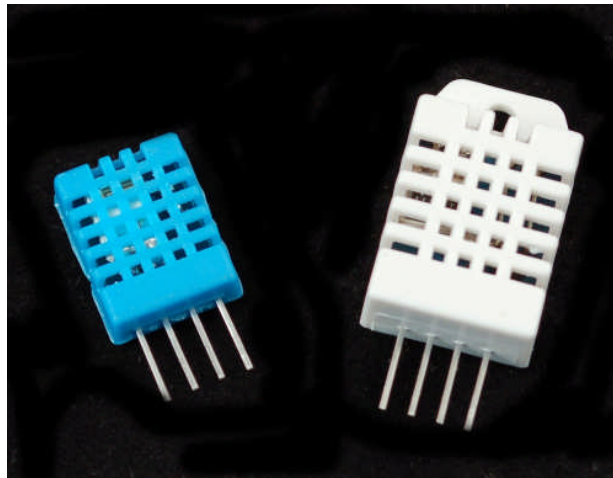


Figure II. 17: DHT22

II.3.3.2. Caractéristiques :

- Alimentation: 3,3 à 6 Vcc
- Consommation maxi: 1,5 mA
- Consommation au repos: 50 μ A
- Plage de mesure:
 - température: -40 à +80 °C
 - humidité: 0 à 100 % RH
- Précision:
 - température: $\pm 0,5$ °C
 - humidité: ± 2 % RH
- Dimensions: 25 x 15 x 9 mm
- Faible coût
- Taux d'échantillonnage ne dépassant pas 0,5 Hz (une fois toutes les 2 secondes)
- 4 broches avec un espacement de 0,1 »

II.3.3.3. câblage

Broche 1 : à gauche à la tension d'alimentation comprise entre 3 et 5V.

Broche 2 : à votre broche d'entrée de données.

Broche 3 : non connecté.

Broche 4 : Masse.[14]

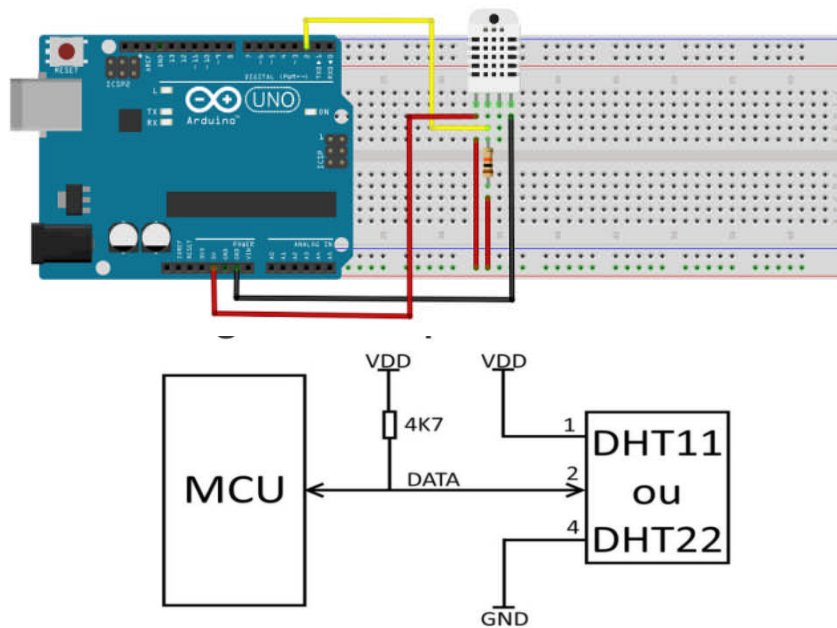


Figure II.18: cablage d'un DHT22

II.4. le Buzzer

Le Buzzer est un composant constitué essentiellement d'une lamelle réagissant à l'effet piézoélectrique. La piézoélectricité est la propriété que possèdent certains minéraux de se déformer lorsqu'ils sont soumis à un champ électrique. Ce phénomène est réversible : si nous déformons ce minéral, il produit de l'énergie électrique.

Dans l'univers Arduino, le Buzzer est principalement utilisé pour émettre un son.



Figure II.19 : le Buzzer.

II.5. Shields DK electronics L298N

Le DK electronics (figure II.16) est un module destiné pour le contrôle de moteurs à courant continu par deux ponts de H (Motor Drive Shields L298N). Il peut contrôler deux moteurs à courant continu ou un moteur pas à pas . Son alimentation peut atteindre jusqu'à 35V.

Il permet de faire tourner les moteurs dans un sens ou l'autre sens sans avoir modifié les branchements.

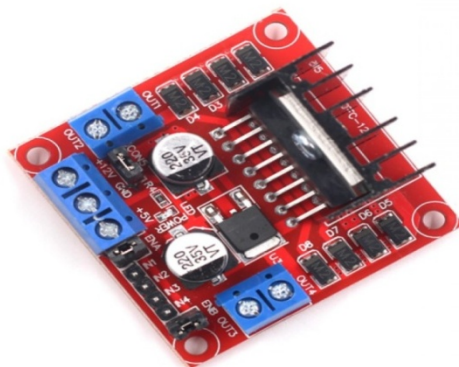


Figure II.20 : Shield DK electronics(L298N).

II.5.1.Caractéristiques de Shield DK électronique

- Compatible avec les cartes Arduino Uno, Mega, Diecimila & Duemilanove.
- Peut piloter 2 moteurs à courant continu DC, ou un moteur pas à pas.
- 2 ponts en H (H-Bridges).
- Pilotage des moteurs à courant continu de 6V à 36V.
- Courant de commande logique : de 0 à 36mA.
- Température de fonctionnement : de -25° à +130°.
- Courant Max (en pointe) : 2A.
- Dimensions : 60mm x 54mm.
- Poids : environs 48g.

II.6.Moteur à courant continu (MCC)

Un moteur est un composant de conversion d'énergie électrique en énergie mécanique.

Les moteurs à courant continu (figure II.17) transforment l'énergie électrique en énergie mécanique de rotation, pour être précis. Mais ils peuvent également servir de générateur d'électricité en convertissant une énergie mécanique de rotation en énergie électrique. C'est le cas par exemple de la dynamo.

❖ Le moteur à courant continu est composé de deux parties principales :

Le rotor (partie qui tourne) et le stator (partie qui ne tourne pas, statique). En électrotechnique le stator s'appelle aussi inducteur et le rotor s'appelle l'induit. Sur la (figure II.19), on peut l'observer au milieu entouré par les aimants bleu et rouge qui constituent le stator le rotor composé de fils de cuivre enroulés sur un support lui-même monté sur un axe. Cet axe, c'est

l'arbre de sortie du moteur. C'est lui qui va transmettre le mouvement à l'ensemble mécanique (pignons, chaîne, actionneur...) qui lui est associé en aval.[8]

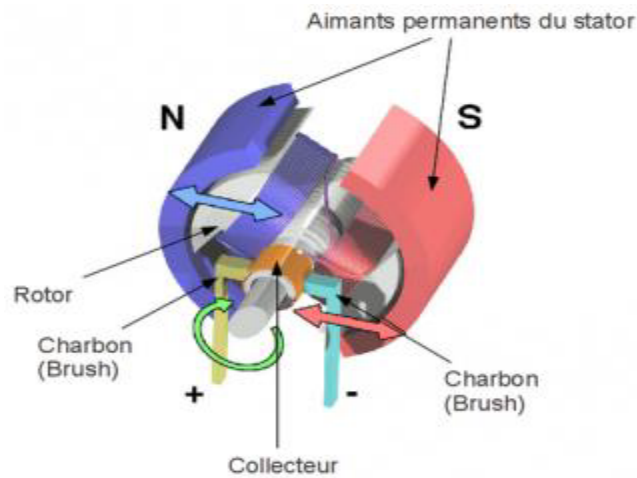


Figure II.21: Moteur à courant continu.

II.7.Servomoteurs

Les servomoteurs servent en principe à actionner les parties mobiles d'un système. Ils sont prévus pour être commandés facilement en position ou en vitesse. En effet, ils sont équipés d'un système d'asservissement basé sur un potentiomètre rotatif qui sert de capteur de rotation.

C'est un ensemble mécanique et électronique comprenant :

- Un moteur à courant continu de petite taille.
- Un réducteur en sortie de ce moteur diminuant la vitesse mais augmentant le couple.
- Un potentiomètre (faisant fonction de diviseur résistif) qui génère une tension variable, proportionnelle à l'angle de l'axe de sortie.
- Un dispositif électronique d'asservissement.
- Un axe dépassant hors du boîtier avec différents bras ou roues de fixation.

Ils sont faciles à utiliser car ils ne nécessitent que :

- Une source de tension continue.
- Une sortie PWM (Pulse Width Modulation) du microcontrôleur de notre robot

II.7.1.Fonctionnement du servomoteur

- Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion plus communément appelés PWM

(Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou RCO (Rapport Cyclique d'Ouverture).

- Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur.[15]
- Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.



Figure II.22 : servomoteur

II.7.2.Caractéristiques :

- ❖ Alimentation : 4,8 à 6V
- ❖ Angle de rotation : 180°
- ❖ Couple : 1,3 kg.cm
- ❖ Vitesse : 0,12 sec/ 60°
- ❖ Dimensions : 23,2 X 12,5 X 22 mm
- ❖ Poids : 9g

II.8.Le module Bluetooth HC-05

Le module Bluetooth est utilisé pour une transmission à distance sans fil

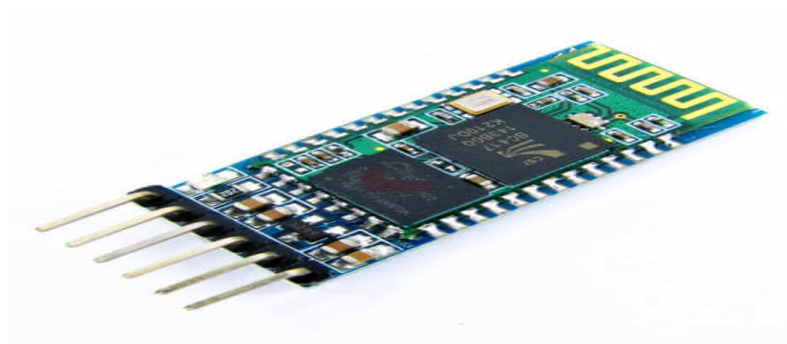


Figure II.23 : module Bluetooth HC-05

II.8.1. Pourquoi choisir le module Bluetooth HC-05

- Consomme moins que le Wifi
- Peu encombrant
- Meilleur débit
- Module moins chère que le Wifi

II.9. L'environnement de programmation Arduino

L'environnement de programmation Arduino appelé IDE est distribué sur le site d'Arduino (compatible avec Windows, Linux et mac). L'interface de l'application Arduino vous permet de créer et éditer un programme (appelé sketch) qui sera compilé puis téléversé sur la carte Arduino. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme téléversé sur la carte. [8]

II.9.1 Fenêtre générale de l'application IDE

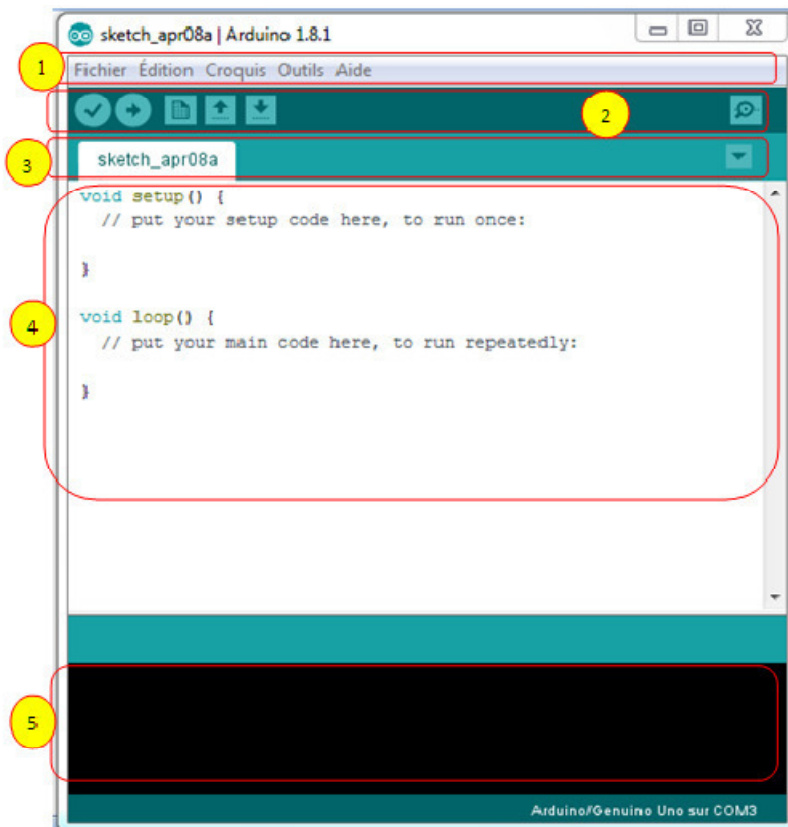


Figure II.24: application Arduino 1.8.1

1. Un menu : permet créer de nouveaux sketches, de les Sauvegarder et gérer les paramètres de communication avec la carte Arduino.

2. Une barre d'actions.
3. Un ou plusieurs onglets correspondant aux sketches .
4. Une fenêtre de programmation .
5. Une console qui affiche les informations et erreurs de compilation et de téléversement du programme.

➤ **La barre d'action**






	<p>Bouton « Verify » (Vérifier) ; il permet de compiler votre programme et de vérifier si des erreurs s'y trouvent. Cette procédure prend un certain temps d'exécution et lorsque est terminée, elle affiche un message de type « Binary sketch size : ... » indiquant la taille du sketch téléversé.</p>
	<p>Pour transmettre le sketche compilé avec succès sur la carte Arduino dans le microcontrôleur.</p>
	<p>Bouton « New » (Nouveau) ; ce bouton permet de créer un nouveau sketch.</p>
	<p>Bouton « Open » (Ouvrir) ; il fait apparaître un menu qui permet d'ouvrir un sketch qui figure dans votre dossier de travail ou des exemples de sketches intégrés au logiciel.</p>
	<p>Bouton « Save » (Sauvegarder) ; il permet de sauvegarder votre sketch.</p>

Tableau [II.4]: Barre d'actions IDE Arduino.

II.10. Logiciel ISIS PROTEUS

Avant de passer à la réalisation, nous devons procéder à des simulations. Cela se fait avec des logiciels de simulations électroniques, ISIS PROTEUS est parmi les logiciels les plus utilisés.

II.10.1. Définition

ISIS PROTEUS est un logiciel de développement et de simulation d'application via un environnement graphique simple et interactif. Ce projet est une initiation à ce logiciel, basé sur un ensemble de composant, dont les résultats doivent être comparés par la suite a ceux obtenus théoriquement et expérimentalement.[16]

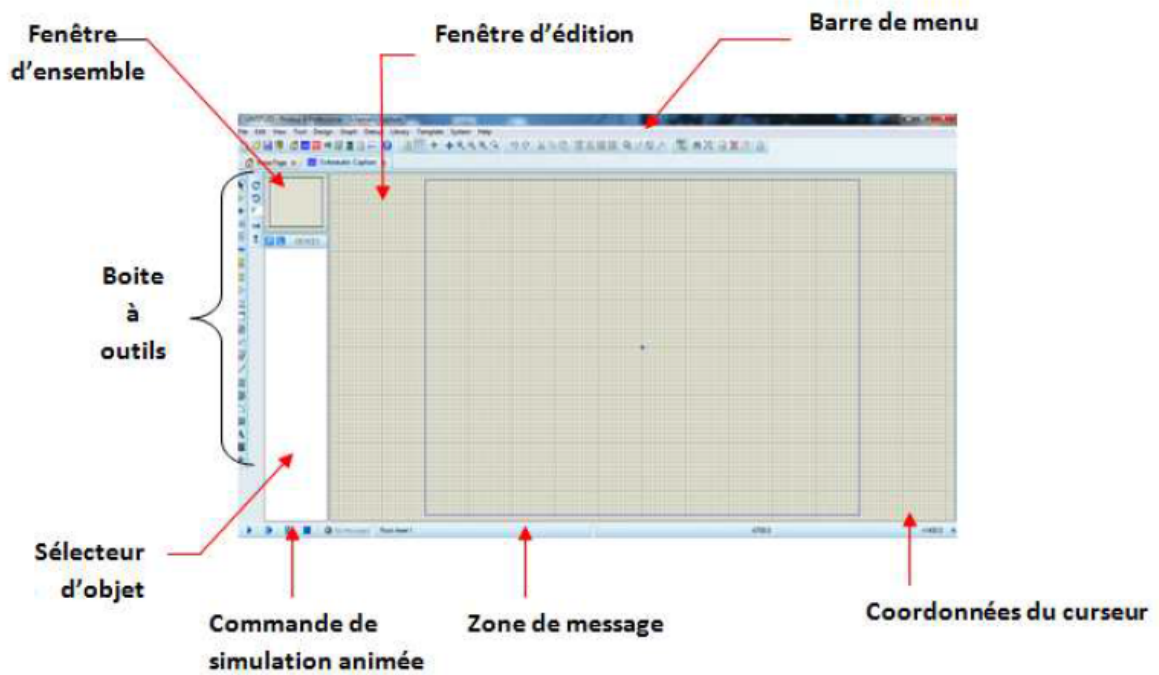


Figure II.25 : Interface d'utilisateur du logiciel proteus

II.10.2.fonctionnement de ISIS PROTEUS

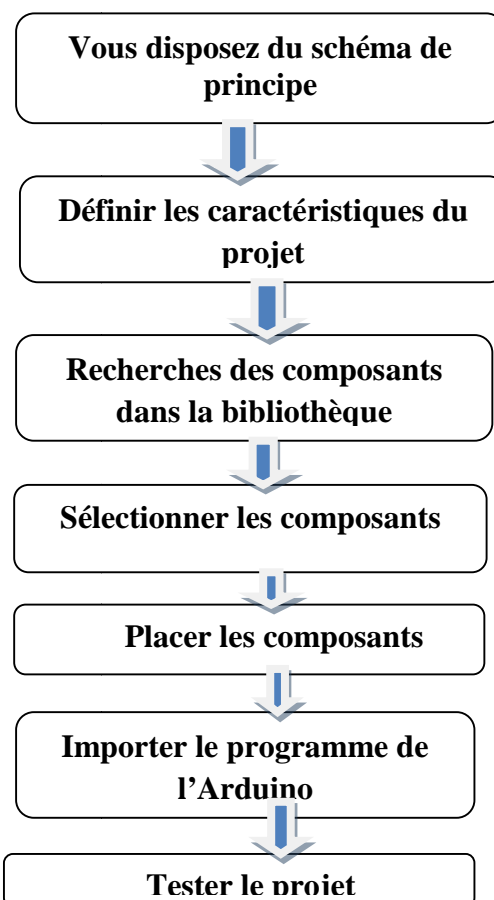


Figure II.26 :schéma de fonctionnement de ISIS Proteus.

II.10.3. Avantages de l'utilisation de ISIS PROTEUS

- logiciel facile et rapide à comprendre et utiliser
- Il permet de tester et vérifier le bon fonctionnement du programme
- Capacité de résoudre des problèmes réels

II.11 .présentation de ESP32-CAM

II.11.1. définition de l'ESP32-CAM

L'ESP32-CAM est une carte de développement ESP-WROOM-32 du fabricant AI Thinker, El **ESP32-CAM** C'est un module que vous pouvez utiliser avec une multitude de projets, et avec Arduino. C'est un module complet avec un microcontrôleur intégré, qui peut le faire fonctionner indépendamment. En plus de la connectivité WiFi + Bluetooth, ce module dispose également d'une caméra vidéo intégrée et d'un emplacement microSD pour le stockage.



Figure II. 27 : Système Esp32-cam

II.11.2.FONCTIONNEMENT

Ce module est facilement programmable via l'IDE Arduino® et permet d'accéder au flux vidéo de la caméra via un réseau local.

- La caméra livrée avec le microcontrôleur ESP32 s'enchâsse dans le connecteur prévu. Il est nécessaire de faire attention lors de la manipulation de la nappe de la caméra, celle-ci est très fragile.
- Un lecteur de carte micro-SD permet le stockage d'images, de vidéos ou de pages HTML.
Carte micro-SD de 4 Go maxi, non incluse.
- Ce microcontrôleur ne dispose pas de convertisseur USB-série intégré.
- Il est nécessaire d'utiliser un convertisseur externe, comme le convertisseur USB-Série FTDI .

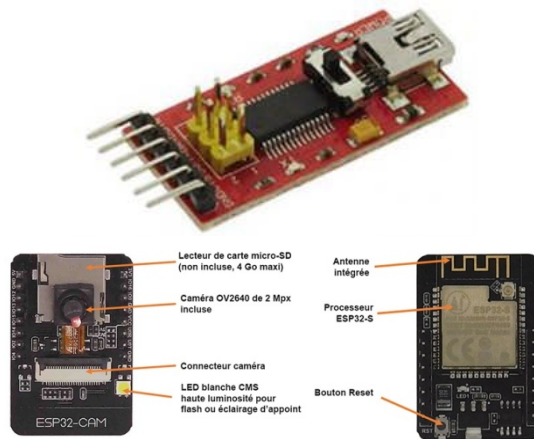
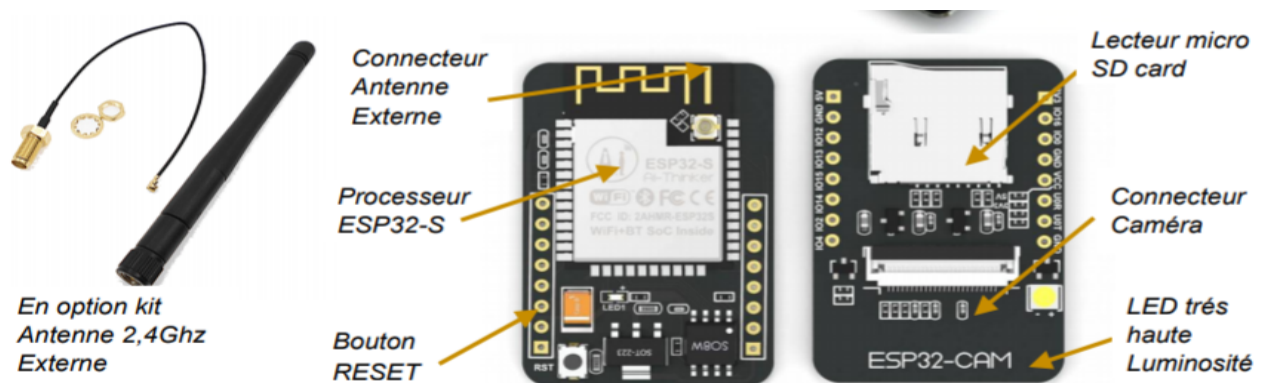


Figure II.28 : principe de fonctionnement

II.11.2 LES COMPOSAN D'ESP32-CAM

La carte ESP32-Cam intègre :

- Un processeur ESP32.
- Une caméra OV2640 (2M pixels) elle consiste à transmettre en WIFI et en direct un flux vidéo et des images.
- Elle est équipée d'un lecteur de cartes microSD qui pourra servir à enregistrer des images des vidéos lorsqu'un événement est détecté (détecteur de présence ou de mouvement par exemple).
- Une LED haute luminosité peut être commandée pour des prises de vues la nuit par exemple. Dans ce cas, il est possible de connecter une photorésistance sur la carte.
- Puelques ports GPIO pour assembler différents capteurs et actionneurs.[17]



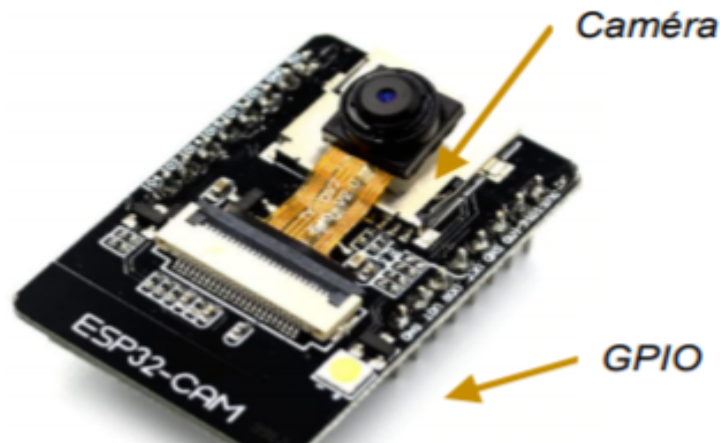


Figure II.29 : les composants d'ESP32-CAM

II.11.3 caractéristiques d'ESP32-CAM

Voici les différentes caractéristiques d'ESP32-CAM

- **connectivité:** Wifi 802.11b / g / n + Bluetooth 4.2 avec BLE. Prend en charge le téléchargement d'images
- via Wifi.
- **Liens:** UART, SPI, I2C et PWM. Il a 9 broches GPIO.
- **Fréquence d'horloge:** jusqu'à 160Mhz.
- **Puissance de calcul du microcontrôleur:** jusqu'à 600 DMIPS.
- **mémoire:** 520 Ko de SRAM + 4 Mo de PSRAM + fente pour carte SD
- **Extras** : dispose de plusieurs modes de veille, d'un micrologiciel pouvant être mis à niveau par OTA et de voyants pour l'utilisation de la mémoire flash intégrée.
- **Appareil photo:** Prend en charge les caméras OV2640 (OV2640 inclus) qui peuvent être livrées dans le pack ou achetées indépendamment et OV7670 avec flash intégré.

Ces types de caméras ont:

- 2 MP sur votre capteur
- Taille de la matrice UXGA 1622 × 1200 px
- Format de sortie YUV422, YUV420, RGB565, RGB555 et compression de données 8 bits.
- Vous pouvez transférer une image entre 15 et 60 FPS.
- CPU 32 bits dual-core basse consommation.
- Prend en charge les interfaces telles que UART / SPI / I2C/PWM / ADC / DAC.

- Lwip et FreeRTOS intégrés.
- Support du mode de fonctionnement STA / AP / STA + AP.
- Support du réseau de distribution en un clic Smart Config / AirKiss.
- Prise en charge de la mise à niveau locale en série et de la mise à niveau du micrologiciel à distance (FOTA).[18]

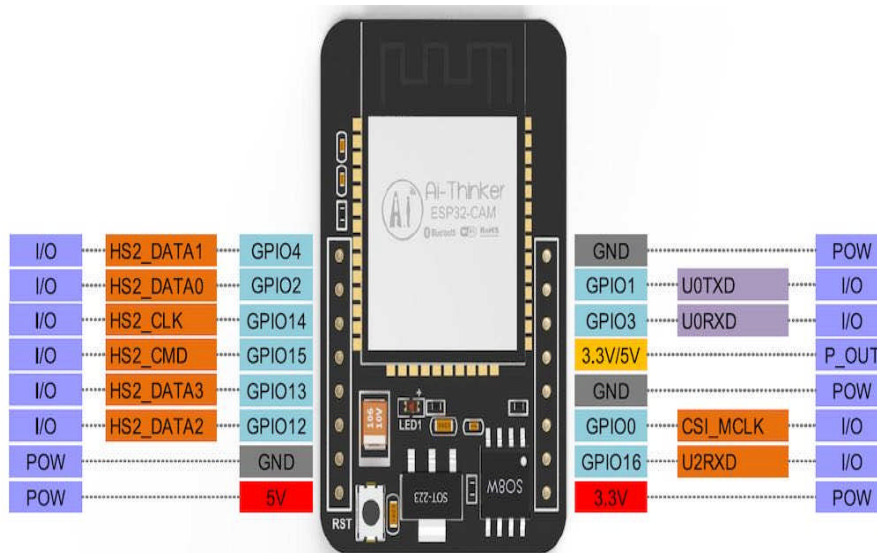


Figure II.30 : la fiche technique d'esp32-cam

II.11.4. LE PRIX

Le module ESP32-CAM n'est pas cher du tout, comme je l'ai dit, pour quelques euros vous pouvez en avoir un. Et vous pouvez facilement le trouver dans certains magasins spécialisés ou sur Amazon.

II.11.5. Installation et Programmation de l'ESP32-CAM

Il existe en effet plusieurs méthodes pour programmer une carte ESP32, mais par soucis d'uniformisation et pour limiter le nombre de logiciels requis dans les tutoriels FabLab, nous utiliserons l'outil IDE Arduino qui est assez simple d'utilisation.

La carte Arduino est très facilement programmable avec ce logiciel, cependant il sera nécessaire de réaliser quelques préparatifs pour pouvoir programmer l'ESP32, notamment installer la carte ESP32 en elle-même.

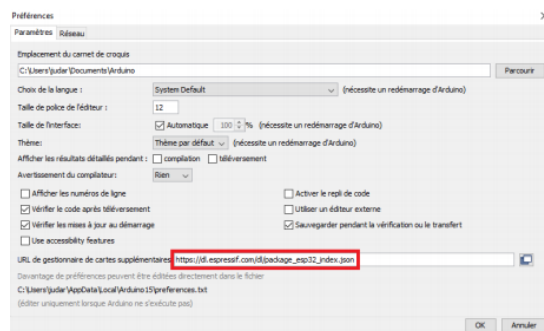
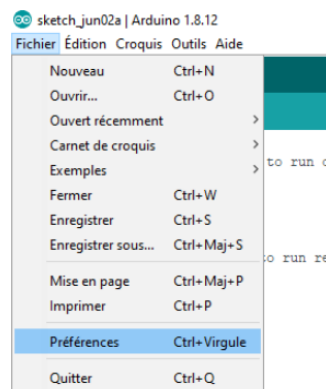
- Les étapes à suivre sont
 - a. Télécharger et installer l'IDE Arduino depuis le site officiel Arduino.cc.

Download the Arduino IDE



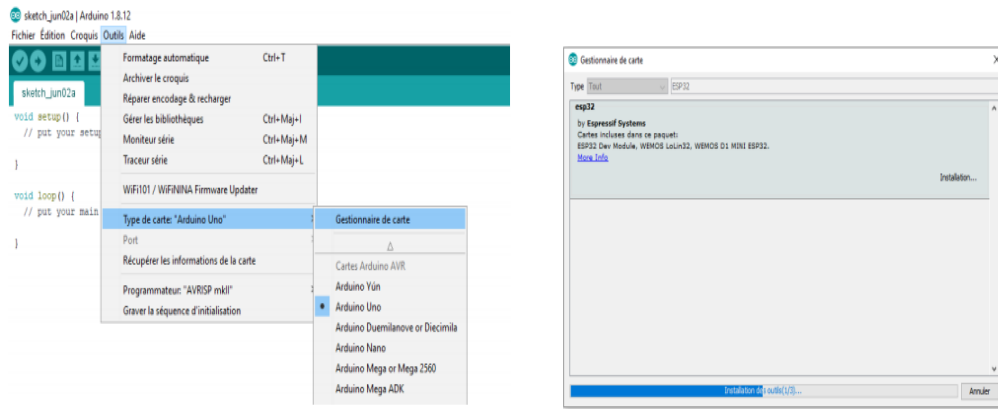
b. Ajouter la carte ESP32 dans la base des cartes Arduino de l'IDE Dans l'IDE Arduino :

- Ouvrir l'onglet Fichier > Préférences .
- Entrer l'adresse suivante dans le champ URL de gestionnaire de cartes supplémentaires : https://dl.espressif.com/dl/package_esp32_index.json



c. Installer la carte ESP32 dans l'IDE Arduino Toujours dans l'IDE Arduino :

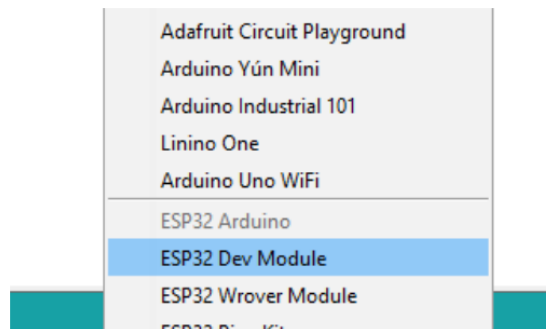
- Ouvrir l'onglet Outil > Type de carte > Gestionnaire de carte.
- Rechercher le paquet ESP32 • Installer le paquet : sélectionnez la dernière version disponible et cliquer sur Installer.



d. Sélectionner la carte ESP32

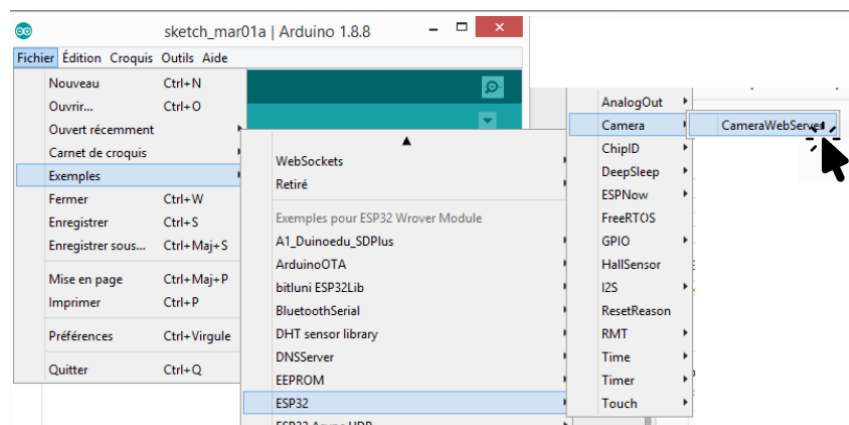
Il ne reste plus qu'à définir la carte ESP32 pour que le programme soit correctement envoyé sur la carte. Toujours dans l'IDE Arduino :

- Ouvrir l'onglet Outil > Type de carte.
- Choisissez alors la carte ESP32 qui correspond à votre version d'ESP32. Le plus souvent ce sera la carte ESP32 Dev Module.



e. Programmé l'ESP32-CAM

Aller dans Fichier-->Exemples-->ESP32-->Caméra-->Ouvrir le fichier CameraWebServer.ino.



f. La dernière étape est de vérifier le programme et puis le téléverser



II.13. Le Châssis

Le châssis (figure II.29) est un dispositif sur laquelle sont posés tous les composants de notre robot, cette petite plateforme à chenilles avec garde au sol réglable dispose de 2 moteurs à courant continu avec encodeur permettant de réaliser facilement un robot mobile. Ce châssis est conçu pour s'adapter facilement avec toutes les extensions telles que la carte de contrôle et les capteurs.

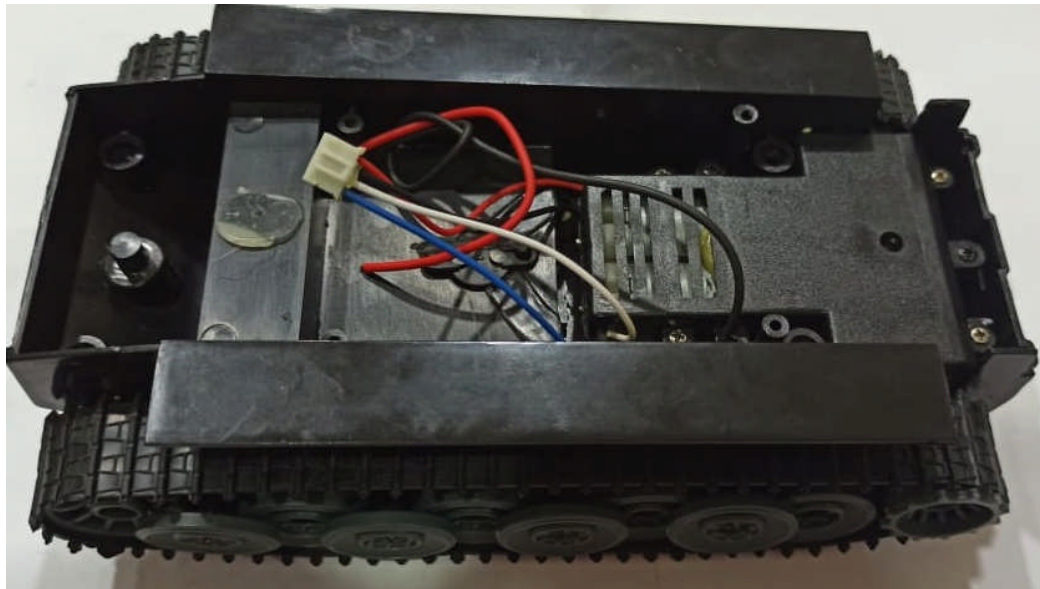


Figure II.31 :le châssis

II.13.1.Caractéristiques

-Motorisation encodeurs	2 moteurs à courant continu
- Tension d'alimentation :	5 V
-Consommation :	1 A
-Couple pour 2 moteurs :	5 kg-cm

Tableau [II.5] : Les caractéristiques du châssis.

II.14. Conclusion

Dans ce chapitre, nous avons présenté les outils matériels utilisés tel que la carte Arduino Uno, les différents capteurs, le moteur à courant continu et le servomoteur. Ainsi que le logiciel Arduino et ISIS PROTEUS .

Chapitre 3

Asservissement Visual d'un robot mobile

III . Introduction

La réalisation d'un système robotique mobile intelligent est liée directement à la puissance d'apprendre ou de s'adapter à son environnement il est nécessaire d'utiliser des capteurs qui fournissent la perception requise de l'environnement pour une prise de décision intelligente , est une caméra pour la réalisation d'un asservissement visuel dans ce chapitre, nous nous intéressons à la navigation par asservissement visuel d'un robot mobile dans un environnement guidé par une caméra placée au dessus du robot.

Dans ce chapitre, nous proposons de réaliser une navigation par asservissement visuel. Pour cela, l'image obtenue par la caméra doit être préalablement traitée afin de fournir les informations visuelles.

III.1. Les systèmes de vision dans la robotique

III.1.1. Les systèmes de vision

En robotique les systèmes de vision sont basés sur l'utilisation d'une caméra CCD (Charge Coupled Device). L'utilisation d'une seule camera ne peut fournir qu'une seule information 2D qu'on appelle vision monoculaire. Pour obtenir des informations 3D, il existe différentes techniques qui sont généralement liées à l'adjonction d'un autre capteur; par exemple: l'utilisation de deux caméra nous permet de reconstruire la structure 3D des objets avec deux images, on l'appelle: la stéréovision. On peut aussi ajouter un système de réflexion de type miroir donnant ainsi la vision omnidirectionnelle. Donc, on peut dire qu'on peut identifier trois techniques: La vision monoculaire, la stéréovision et La vision omnidirectionnelle.

a. La Vision Monoculaire

La vision monoculaire utilise une seule caméra qui observe directement la scène. L'inconvénient de ce système est: son champ visuel est limité et ne permet pas de restituer la profondeur de la scène. Il ne fournit qu'une seule représentation bidimensionnelle de l'image et de l'espace tridimensionnel, ce qui n'est pas suffisant. La profondeur est déterminée à partir d'une ou plusieurs images. En vision par ordinateur, ces techniques de reconstruction 3D à partir d'images sont appelées techniques de « Shape-From-X », où X peut être « Stéréo », « Motion », « Shading », etc.

b. La Stéréovision

La stéréovision consiste à utiliser deux caméra séparées l'une de l'autre d'une certaine distance connue et observant la même scène .Lorsqu'on utilise deux caméras, c'est le système binoculaire, il est suffisant pour reconstruire en 3D la scène observée ou retrouver

l'information de la profondeur par triangulation qui peut être utilisée pour l'évitement d'obstacles ou pour la cartographie. [20]

c. La Vision Omnidirectionnelle

Le terme de la vision omnidirectionnelle se réfère au capteur de vision avec un très large champ de vue, c'est-à-dire un capteur de champ de vue de 360° . Comme il existe un autre terme pour un champ de vue large c'est la vision panoramique. Ces deux termes sont généralement employés indifféremment alors qu'il existe une différence entre eux.

- La vision omnidirectionnelle est le procédé de vision avec un champ de vue de 360° horizontale avec un champ de vision vertical qui peut être limité ou non, donc son champ de vue doit être sphérique, et permet d'observer le monde à partir de son centre.

La vision panoramique est une réduction du premier procédé, son champ de vue est limité. Il est de 360° horizontale avec un champ de vue vertical limité. Donc elle ne couvre pas tout l'environnement dans un autre sens.

Plusieurs techniques ont été développées pour augmenter le champ de vue panoramique et omnidirectionnel que l'on peut classer en trois catégories. [21]

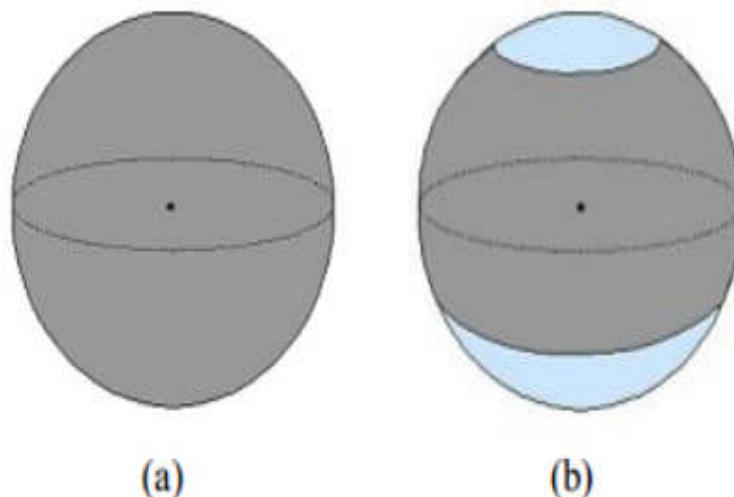


Figure III.32 :champ de vue des système de vision

(a) Vision panoramique. (b) vision omnidirectionnelle

❖ Fusion de plusieurs images

La première catégorie est basée sur l'utilisation de plusieurs images pour former une image panoramique. Cependant ce système pose plusieurs problèmes en pratique. Deux techniques peuvent être mises en œuvre pour l'acquisition des images:

- La rotation d'une caméra autour d'un axe;
- Utilisation de plusieurs caméras.

• Rotation d'une caméra

La première technique consiste à effectuer une acquisition d'une image par rotation d'une caméra autour d'un axe vertical pour acquérir l'image panoramique. Aussi, il est possible d'utiliser une caméra matricielle. [22] La caméra utilisée dans un tel système est en général linéaire puisqu'il suffit alors de concaténer les lignes verticales pour former l'image panoramique.

• Systèmes multi-caméras

La deuxième technique est basée sur l'utilisation de plusieurs caméras de manière à obtenir une acquisition du panorama instantanément, contrairement à la première technique.

L'obtention d'une haute résolution est l'avantage de ces systèmes "multi-caméras" mais il a des inconvénient

- Il est très cher.
- La difficulté de synchroniser toutes les caméras.
- L'importance du flux de données à traiter.

❖ Utilisation d'un objectif grand angle:

Le grand angle, appelé aussi grand angulaire, est un objectif de courte focale. Certains objectifs permettant d'obtenir un grand angle sont les objectifs type "Fish-eye" (œil de poisson). Une caméra "fish-eye" est une caméra catadioptrique possédant une focale très courte et une association d'un miroir de révolution placé devant la caméra pour agrandir le champ de vue.

❖ Utilisation d'un capteur catadioptrique:

Les caméras qui utilisent uniquement l'effet de réfraction de la lentille de refléter la lumière sont appelés caméras dioptriques (la dioptrie est la science de éléments de réfraction). Les caméras qui utilisent les effets combinés de la réflexion sur un miroir et de réfraction d'une lentille sont appelés les caméras catadioptriques (à partir de catoptrique, c'est à dire la science des surfaces de réflexion et dioptrique). Un capteur catadioptrique est composé d'une caméra et d'un miroir ayant en général un axe de révolution.



Figure III.33 : Exemple de capteur catadioptrique.

III.2. Modélisation de la Caméra

III.2.1. Conception

Une caméra doit réaliser une transformation ponctuelle qui fait passer d'un point physique de l'espace réel 3D à un point 2D sur le plan image. Ce qui revient à une transformation mathématique de R^3 vers R^2 .

Il existe plusieurs modèles dans la modélisation de la formation des images numériques. Notre étude prend comme modèle celui du sténopé : appelé également le modèle perspectif (pin-hole en anglais) : c'est un dispositif optique et le plus utilisé dans la vision par ordinateur. Ce modèle permet d'établir une relation entre un point de coordonnées 3D de la scène observée et sa projection dans l'image en 2D.

III.2.2- Composants et la modalisation d'une caméra

Une caméra se compose d'une boîte dont l'une de ses faces est percée d'un trou minuscule qui laisse entrer la lumière comme indique. L'image vient se former sur la face opposée au trou et celle-ci peut être capturée en y plaçant un support photosensible (papier photographique).

Et sa modalisation consiste à présenter la projection du point de l'espace 3D à un point 2D sur le plan image par trois transformations élémentaires successives : la transformation rigide, la projection perspective et la transformation affine.

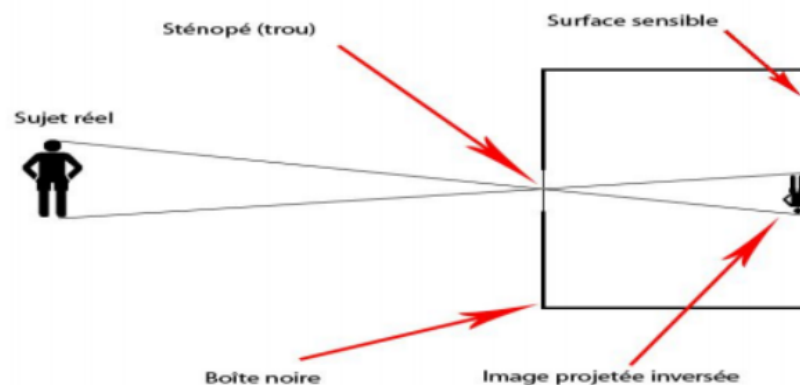


Figure III.34 : formation d'une image dans un sténopé

III.3.L'asservissement visuel

III.3.1. Algorithme de poursuite d'une trajectoire

III.3.1.a. L'acquisition et traitement d'image

L'acquisition est la première étape pour le traitement et l'analyse de l'image, pour avoir une bonne prise d'images on doit s'assurer d'être entouré de meilleures conditions. Le système d'acquisition est composé par :

- Une caméra couleur de type Esp32-cam.
- D'une base mobile du robot.
- Un téléphone.

L'image acquise est représentée sous forme des pixels contenant un très grand nombre de données. Étant donné que l'image obtenue est contaminée par le bruit, il s'avère nécessaire de faire un prétraitement pour s'en débarrasser et améliorer ainsi sa qualité.

III.3.1.b. Extraction des caractéristiques

Avec l'étape de prétraitement, on obtient une image dotée de bonne qualité qui permet de faire une extraction des caractéristiques qui serviront à identifier la trajectoire à poursuivre.

III.3.1.c. Algorithme de traitement d'image

La segmentation de l'image est la base de tous les systèmes de vision pour l'extraction des caractéristiques ou d'une carte de l'environnement. Pour détecter et caractériser les zones d'intérêts il faut traiter l'image.

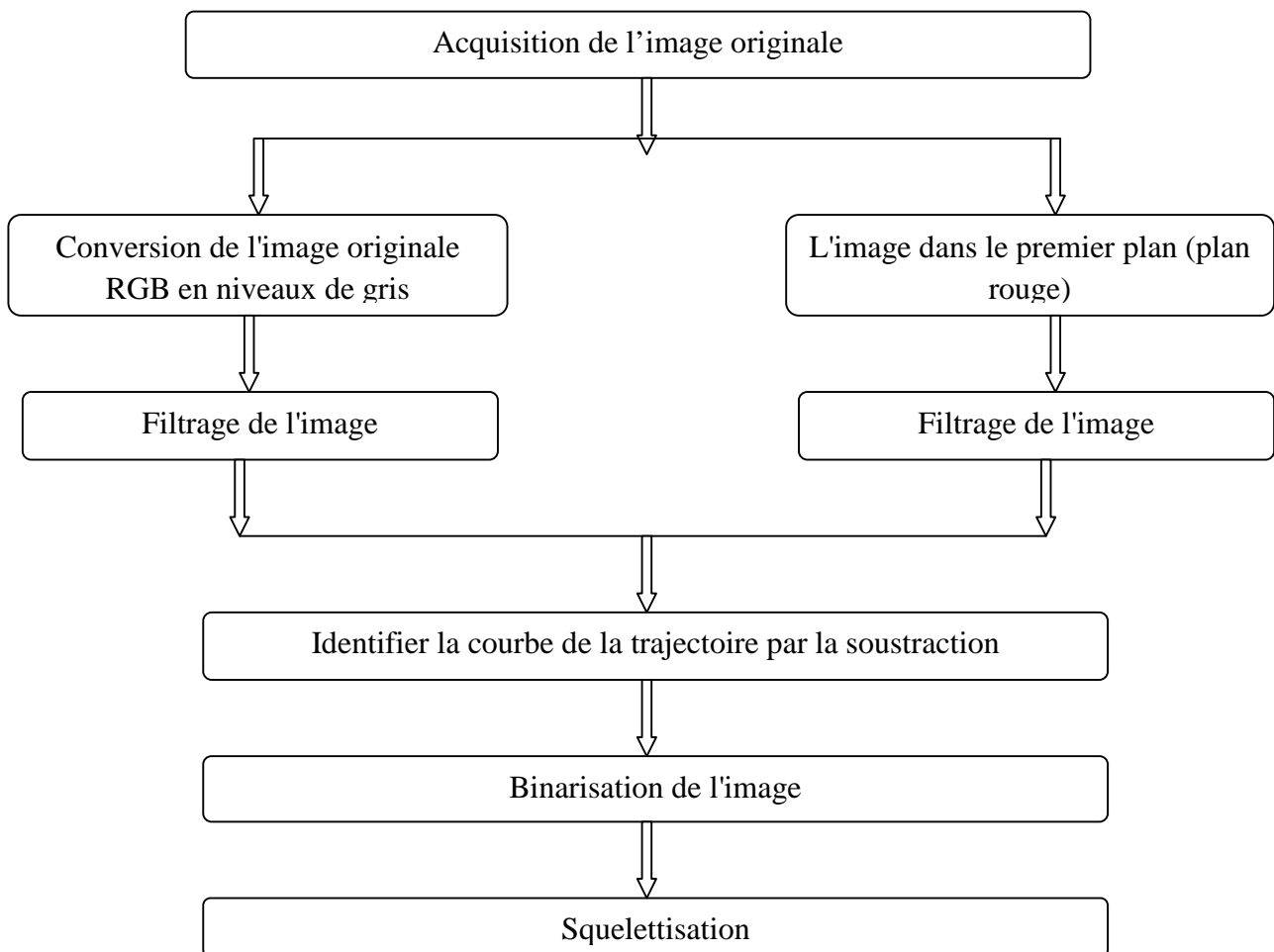


Figure III.35 : algorithme de traitement d'image

❖ Convertir l'image RGB en niveaux de gris

Chaque pixel de l'image RGB peut être décomposé à un ensemble de trois valeurs de 8 bits, correspondant à la quantité (intensité) de rouge, vert et bleu (en anglais : Red, Green, Blue). Chaque composante de couleur de l'image est désigné par un plan image ou d'un canal. Les écrans 16 millions de couleurs sont des écrans RVB à 8 bits, soit 256 couleurs par teinte de

base La réduction de chaque pixel a une valeur unique de l'ensemble original de trois valeurs rétrécit le nombre d'opérations par les deux tiers et simplifie encore le traitement de l'image. Ce processus est connu comme la conversion en niveaux de gris, aussi la résultante des pixels de l'image sera stockée sous un numéro unique de 8 bits, souvent représenté par une valeur de gris entre le noir et blanc.

Les images en niveaux de gris sont très utilisables, parce que les images en niveaux de gris sont tout à fait suffisantes pour de nombreuses tâches et il n'y a donc pas besoin d'utiliser des images couleurs plus complexes et plus difficiles. Il existe trois méthodes pour convertir une image RVB en niveaux de gris. Un pixel gris a ses trois valeurs RVB identiques. Une méthode simple pour convertir une image couleur en niveau de gris peut être calculé par la moyenne des trois composantes RVB et d'utiliser cette valeur moyenne pour chacune des composantes:

$$\text{Gris} = \frac{R+V+B}{3} \quad (\text{III.1})$$

Comme existe d'autre formule :

$$\text{Gris} = 0.299*R + 0.587*V + 0.114*B \quad (\text{III.2})$$

$$\text{Gris} = 0.212*R + 0.715*G + 0.072*B \quad (\text{III.3})$$

A noter que la formule utilisée par Matlab est celle donnée par l'expression (III.2).

❖ Filtrage de l'image

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites). Le bruit est un ensemble de pixels qui ne font pas partie de la scène à traiter. Il est injecté par plusieurs sources et pour s'en débarrasser on a besoin de faire un filtrage avant tout traitement de l'image.

Plusieurs méthodes de filtrage ont été développées dans la littérature suivant le type et l'intensité du bruit présent dans l'image. Les plus simples des méthodes sont basées sur le filtrage linéaire stationnaire qui est au fait invariant par translation comme le filtre moyennneur, filtre smooth. Mais les limitations de ces techniques (en particulier leur mauvaise conservation des transitions) a conduit au développement des filtres "non-linéaire" pour pallier aux insuffisances des filtres linéaires : principalement la mauvaise conservation des contours [filtres]. Ils ont le défaut d'infliger des déformations irréversibles à l'image. Les

filtres non linéaires sont basés sur des bases mathématiques ou empiriques différentes comme: le filtre nagao, FAS ou le filtre médian que nous avons choisi pour notre application.

❖ Filtre médian

Le filtre médian dit également filtre spatial à fenêtre glissante. Il est particulièrement efficace contre le bruit dans des images à niveaux de gris, il remplace les valeurs centrales de chaque fenêtre par la valeur médiane de toutes les valeurs de pixels dans la fenêtre. Ainsi, si plusieurs pixels voisins sont bruités, on peut corriger le pixel courant. Ce filtre induit cependant un lissage puisque même des pixels corrects peuvent être modifiés. Par contre, ce filtre est couteux en terme de calculs car il nécessite d'effectuer un tri des voisins pour chaque pixel. Le noyau est généralement carré, mais peut être de n'importe quelle forme. Comme illustration, on prend l'exemple suivant d'un filtre médian d'une fenêtre d'analyse 3x3.

5	3	6
2	11	9
8	4	7

Valeurs non filtrés

Les niveaux de gris sont classés par ordre croissant dans l'ordre croissant : 2, 3, 4, 5, 6, 7, 8, 9, 11

La valeur médiane 6 est choisie

*	*	*
*	6	*
*	*	*

Filtre médian

La valeur de centre (précédemment 11) est remplacé par la médiane de l'ensemble des neuf valeurs : 6

❖ Identifier la courbe de la trajectoire

Cette étape requise pour arriver à détecter la trajectoire désirée consiste à rechercher les couleurs des objets dans l'image. Elle permet d'identifier des objets qui se trouvent dans l'environnement à partir de leur couleur et de supprimer les éléments extérieurs qui ne sont pas importants. La technique utilisée pour isoler la trajectoire est l'opérateur de soustraction.

L'opérateur de soustraction prend deux images en entrée et produit en sortie une troisième image dont les valeurs des pixels sont simplement les valeurs des pixels de la première image moins ceux correspondants à la deuxième image. Il est également souvent possible d'utiliser une seule image comme entrée et soustraire une valeur constante à tous les pixels. Par ailleurs, on peut trouver certaines versions où l'opérateur génère une sortie résultante de la différence entre les valeurs absolues des pixels plutôt qu'une sortie signée.

- La soustraction des deux images est effectuée sans détour en un seul passage. Les valeurs des pixels de sortie sont données par :

$$Q(i,j)=P1(i,j)-P2(i,j) \quad \text{(III.4).}$$

- Ou si l'opérateur calcule les différences absolues entre les deux images d'entrée :

$$Q(i,j)= |P1(i,j)-P2(i,j)| \quad \text{(III.5).}$$

- Ou si elle est simplement souhaitable de soustraire une valeur constante C à partir d'une seule image :

$$Q(i,j)=P1(i,j)-C \quad \text{(III.6).}$$

Si les valeurs des pixels dans les images d'entrées sont des vecteurs plutôt que des valeurs scalaires (par exemple les images en couleurs), alors les composantes individuelles (par exemple les composantes rouge, vert et bleu) sont simplement soustraites séparément pour produire la valeur de sortie.

Lorsque les valeurs des pixels de la sortie sont négatives, on peut avoir plusieurs implémentations de l'opérateur. Certains peuvent fonctionner avec des formats d'image qui soutiennent les pixels de valeurs négatives. La façon avec laquelle ils sont affichés est déterminée par la carte des couleurs de l'écran "colormap". Si le format de l'image ne supporte pas les nombres négatifs alors souvent ces pixels sont mis à zéro (c'est-à-dire typiquement noir). Sinon, l'opérateur peut traiter les valeurs négatives, de sorte que par exemple (-30) apparaît à la sortie 226 (en supposant que les valeurs sont codés en 8 bits).

Dans notre cas les deux images d'entrée sont l'image dans le premier plan (le plan rouge) et l'image originale en niveau de gris, le format de l'image ne supporte pas les nombres négatifs donc si la différence entre les pixels est inférieure à 0 ces valeurs sont remplacées par des valeurs en niveaux de gris codé en 8 bits.

❖ Binarisation de l'image

Les images binaires sont des images dont les pixels ne peuvent avoir que deux valeurs d'intensité possibles. Ils sont normalement affichés en noir et blanc. Numériquement, les deux valeurs sont souvent 0 pour le noir, et 1 ou 255 pour le blanc.

Le principe de « binariser » une image est de comparer chaque pixel de l'image à un seuil et affecter la valeur noire à ce pixel si la valeur d'origine est inférieure au seuil ou bien la valeur blanche dans le cas contraire. La dynamique de l'image est alors réduite à deux luminosités.

❖ Squelettisation

La squelettisation est une transformation largement utilisée dans les champs d'analyse des images et de reconnaissance de forme. La squelettisation dans le plan désigne un processus qui transforme un objet 2D en une représentation d'une ligne 2D. Le concept de squelette a été introduit par Blum en 1961, sous le nom de "la transformation d'axe médian".

L'idée de la squelettisation est de réduire un objet à une représentation de moindre dimension qui capte les principales caractéristiques de l'objet original. Les squelettes peuvent être utilisés pour visualiser des aspects importants d'un objet.

Le squelette d'une forme est défini comme l'ensemble des points dont la distance au point le plus proche du contour est localement maximum [23]. Ainsi, le squelette représente la ligne centrée dans la forme. Par exemple, le squelette d'un cercle est le centre du cercle et le squelette d'un carré est donné par ses diagonales.

Différentes méthodes peuvent être utilisées pour définir un squelette. Nous allons utiliser la squelettisation basée sur l'amincissement. Elle est donnée par une méthode bien connue qui s'appelle la morphologie mathématique. C'est une discipline basée sur la théorie des ensembles.

- L'opérateur morphomathématique

La morphologie mathématique est une théorie de traitement non linéaire de l'information apparue en France dans les années 60 [G. Matheron & J. Serra, Ecole des Mines de Paris], et qui est aujourd'hui très largement utilisée en analyse d'images. Contrairement au traitement linéaire des images, la morphologie mathématique ne s'appuie pas sur le traitement du signal,

mais repose sur la théorie des ensembles, ce qui en fait une discipline relativement « autocontenue » et formant un tout cohérent [27].

La Morphologie mathématique est un outil utile pour la segmentation et le traitement d'image. Initialement appliqués sur les images en noir et blanc par Matheron et Serra en 1965, elle a été ensuite étendue à des images en niveaux de gris par Dougherty en 1978. Pour l'appliquer à des images en couleurs, il suffit alors de traiter chaque couleur séparément.

La morphologie mathématique est un outil pour extraire les composants utiles dans la représentation et la description de la région de forme image, tels que les frontières, les squelettes etc. La langue de la morphologie mathématique est la théorie des ensembles. Elle utilise un ensemble de centre x , de géométrie et de taille connues, appelé élément structurant, c'est à dire des configurations élémentaires de pixels à rechercher dans l'image.

En raison de la complexité de la morphologie à l'échelle de couleur, l'image couleur est transformée en une image binaire par la segmentation. Les opérateurs dans le domaine de la morphologie binaire sont la dilatation, l'érosion et deux opérations complémentaires : la fermeture et l'ouverture sont utilisées pour le prétraitement des données.

- **Éléments structurant**

Un élément structurant est tout simplement est une image binaire ou un masque binaire (constitué de pixels blancs et noirs) qui nous permet de définir arbitrairement des structures de voisinage que l'on souhaite.

- **Squelettisation basée sur l'amincissement**

L'amincissement ou « thinning » est une méthode qui consiste à éplucher la forme jusqu'à obtenir un ensemble de points connectés d'un pixel qui conserve la topologie de la forme : le squelette [24]. Donc elle consiste à réduire une forme en un ensemble de courbes centrées dans la forme d'origine. L'importance de la squelettisation vient du fait que dans le processus de perception visuelle.

L'application de l'amincissement a été introduit par Harry Blum dans les années soixante, de but de créer un nouveau descripteur de formes. La squelettisation sert à la reconnaissance de formes, la modélisation des solides pour la conception et la manipulation de formes, l'organisation de nuages de points, la planification de trajectoire, la détection de réseaux routiers, les animations, etc.

Plusieurs algorithmes d'amincissement sont proposés dans la littérature, L'algorithme d'amincissement que nous avons adopté est de Lam, Seong-Whan Lee, et Ching Y. Suen [25] qui proposent une implémentation morphologique de leur algorithme d'amincissement.

III.3.2. Résultats de simulations de traitement d'image



Figure III.36 :l'image originale



figure III.37 :l'emale en niveaux de gris



Figure III.38 :le plan rouge de l'image



figure III.39 :identification de la trajectoire



Figure III.40 :l'image après filtrage et binarisation

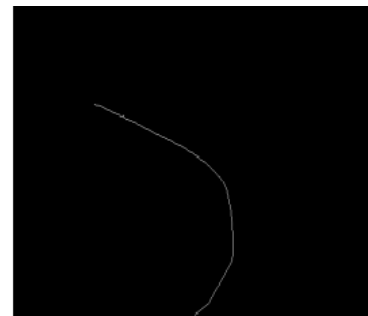


figure III.41 : la squelettisation de trajectoire

III.3.3.Choix des primitives visuelles

Lors d'une navigation dans un environnement en milieu naturel, nous cherchons à caractériser l'environnement à partir d'informations visuelles indépendantes du zoom, de la résolution du capteur, du cadrage, de l'angle d'observation et de l'exposition. Les informations visuelles sont alors des points. Bien que nous utilisons régulièrement des amers

artificiels, notre choix pour réaliser une navigation visuelle se porte sur des primitives de type point.[26]

III .4.CONCLUSION

Dans ce chapitre, nous avons présenté des généralités sur les systèmes de vision en robotique, sur les travaux de recherche effectués dans ce domaine pendant la dernière décennie, on a présenté le calibrage et la modélisation de la caméra utilisée à cet effet. Ce calibrage est nécessaire pour obtenir les paramètres intrinsèques et extrinsèques utiles à l'utilisation de la caméra. La modélisation de la caméra est basée sur le modèle "sténopé" qui est le plus employé dans la vision par ordinateur. Il permet en effet de transformer un point 3D de l'espace à un point 2D dans le plan image.et pour finirai ce chapitre nous avons proposé une méthode de poursuite d'une trajectoire par vision monoculaire. Les coordonnées de la trajectoire désirée sont obtenues par échantillonnage de la trajectoire désirée. Cette trajectoire est obtenue après avoir appliqué un traitement d'image adéquat qui passe d'abord par une acquisition de l'image, filtrage. Comme nous avons un fait un système de vision en robotique et comme nous avons utilise une application prit donc nous somme pas oblige de faire le chapitre de vision par ce que il est prit.

IV. Introduction

Dans les domaines de la robotique et des objets connectés, une des fonctionnalités les plus intéressantes à mettre en œuvre est le pilotage à distance d'un projet électronique. Que ce soit pour afficher des mesures provenant ou bien pour piloter un drone, une solution possible est de développer une application pour Smartphone et de communiquer avec un microcontrôleur en utilisant une liaison Bluetooth. Cette Application peut être rapidement créée en utilisant le logiciel en ligne du MIT, App Inventor 2 (AI2).

On commence d'abord par la présentation de l'application.

IV.1.L'application Android

IV.1.1.Présentation Android

Android est un système d'exploitation libre qui utilise les téléphones mobiles modernes (Smartphones), il est basé sur Linux. Android est détenue par Google.



Figure IV.42 : logo Android

IV.1.2. Présentation de MIT App Inventor

- MIT App Inventor (appinventor.mit.edu) est une application Cloud simplifiée le développement d'application Android en utilisant une interface graphique proche de Scratch.

Avec MIT App Inventor permet de créer facilement des interfaces graphique Android pour le contrôler des applications robotiques comme une voiture, un bras, un robot..ect.

- App inventor est un environnement de développement intégré qui permet de créer des applications destinées à des systèmes équipés de plateforme Android tels que les téléphones et les tablettes.
- A l'origine le projet APP INVENTOR a été créé par le Google Labs puis abandonné. Aujourd'hui ce concept est repris par le MIT (Massachusetts Institute of Technology). Le téléchargement et l'utilisation de APP Inventor nécessite un compte Google (gmail). [19]

L'IDE est formé de deux composantes logicielles :

- Un site Web permettant de créer l'interface de la future application (aspect graphique)
- Un programme résidant sur l'ordinateur, qui permet par association des blocs liés aux objets précédemment définis dans l'espace web, de créer le comportement de l'application.



Figure IV.43 :MIT APP INVENTO

IV.2.Matériel

- Ordinateur
- Smartphone Android avec une application de lecture de code QR
- Arduino UNO
- Bluetooth HC-06.
- Câble Dupont F/M

IV.3.Création de l'application AI2

IV.3.1.Les étapes de création de l'application

1. copier le lien de suivant sur la barre de recherche du navigateur Google :
<http://appinventor.mit.edu/>.
2. Cliquer sur Create apps ! (créer l'application).

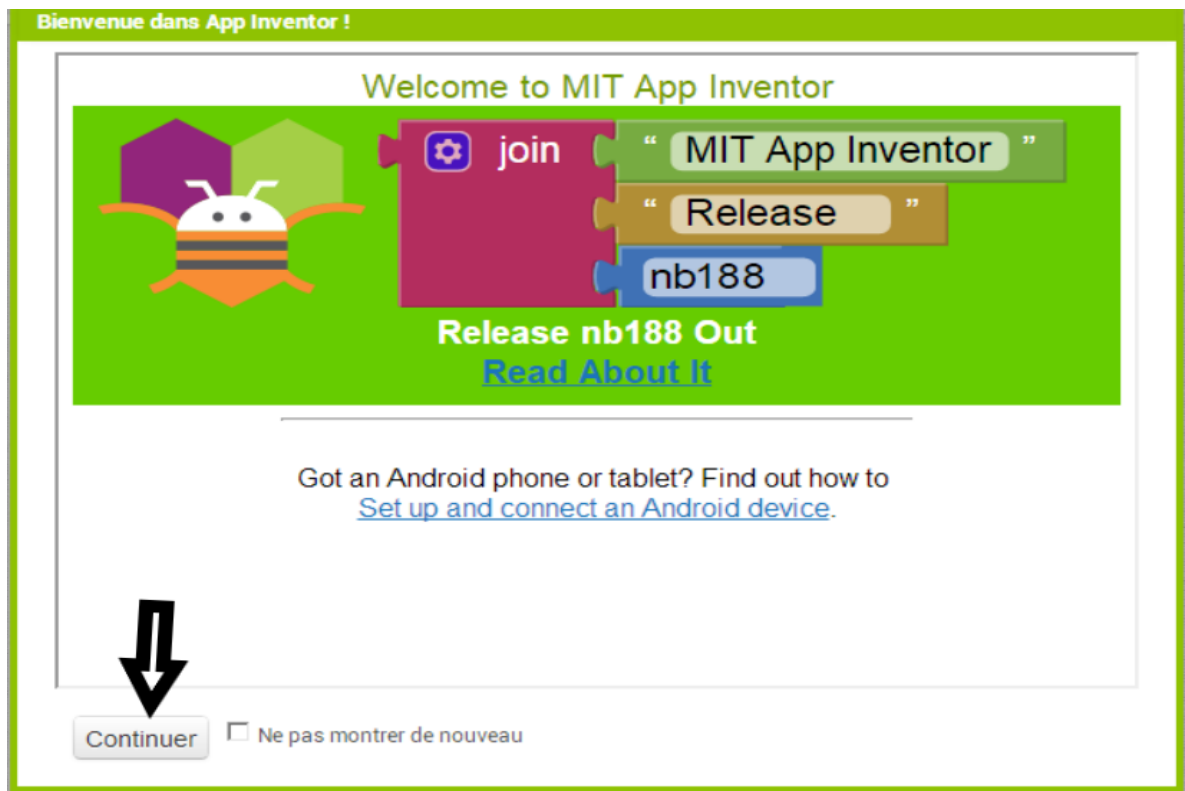
The screenshot shows the MIT App Inventor website. At the top left is the MIT App Inventor logo. To its right is a navigation menu with links: 'Create Apps!' (highlighted with a red arrow), 'About', 'Educators', 'News', 'Resources', 'Blogs', and 'Donate'. A search bar is on the far right. Below the navigation is a large banner with the text: 'With MIT App Inventor, anyone can build apps with global impact'. Underneath the banner is a statistics table:

Active Users	Active Users	Active Users	Registered	Countries:	Apps Built:
today:	this week:	this month:	Users:	195	34.0M
62.6K	242.3K	803.8K	8.2M		

Below the statistics is a news snippet with a 'join' button and a '2020' tag. The text reads: 'The finalists for the MIT App Inventor Hackathon 2020, A Global App Hackathon for Good, have been announced.'

Figure IV.44 :site de MIT APP INVENTOR

3. Cliquez sur "Continue" pour sortir de l'écran d'accueil.



4. Et pour crée un nouveau projet nous avons deux méthodes

- Cliquez sur start new project puis nommez ce projet, tapez le nom du projet (les tirets bas sont autorisés mais pas les espaces) et cliquez sur ok.

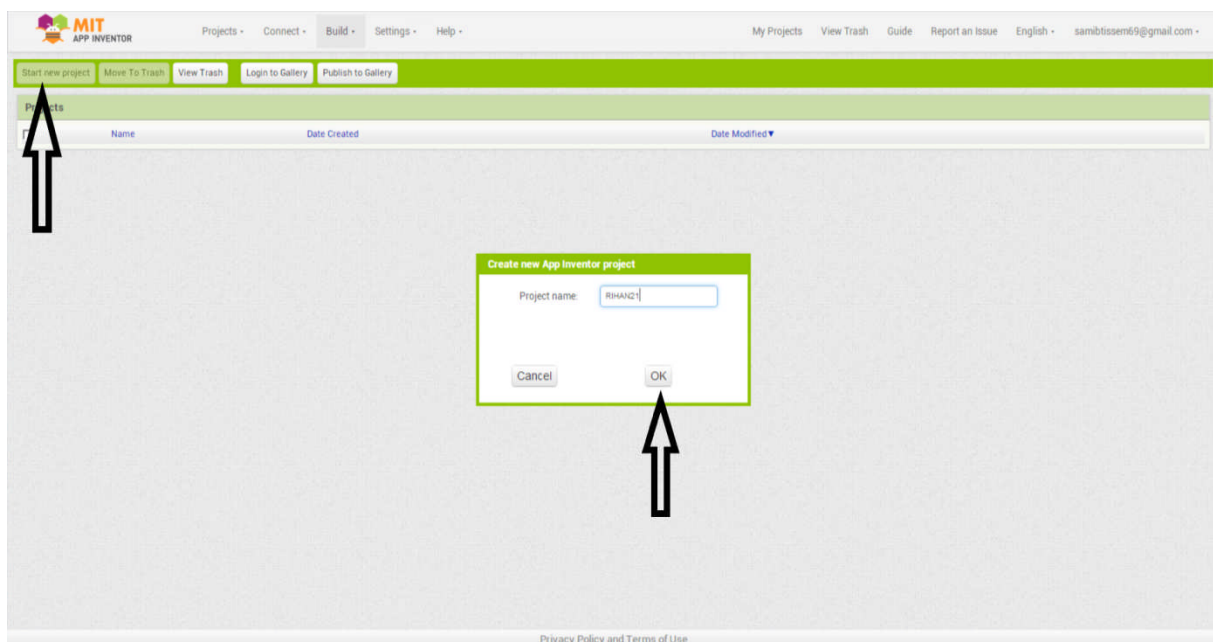


Figure IV.45 : la fenêtre pour nome le projet

- Cliquer sur projects puis sur start new project (créer un nouveau projet)

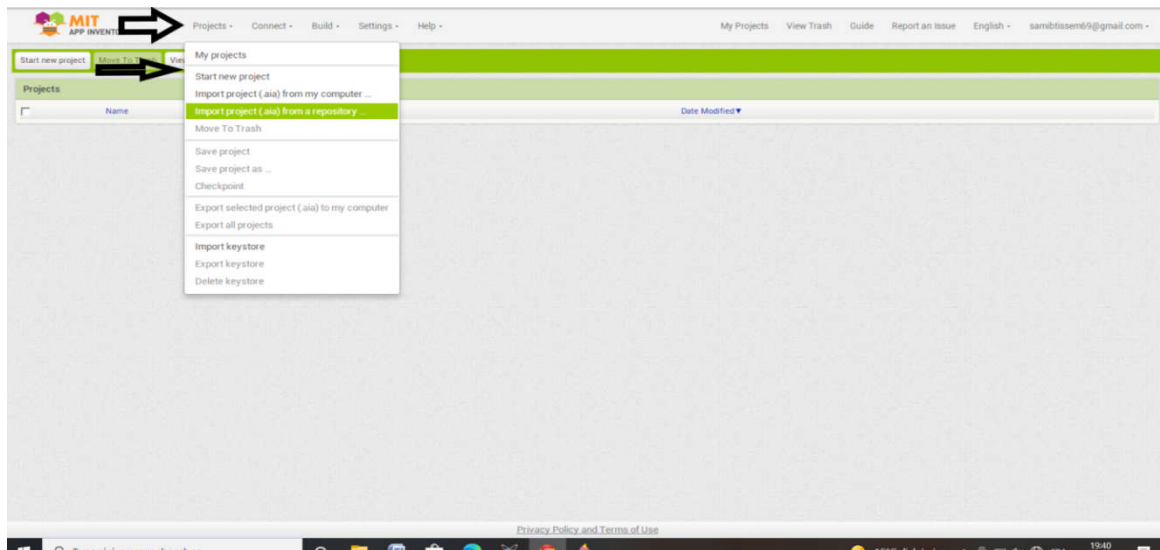


Figure IV.46 : fenêtre des projets.

5. Donner un nom au projet
6. Nous sommes maintenant dans l'espace Designer où il nous est possible d'afficher l'interface (interface utilisateur) de notre appli. Avec Design Windows, ou "Designer", nous pouvons visualiser l'apparence de notre application et sélectionner les composants que nous doit avoir. dans cette interface on peut aussi choisir des options pour l'interface utilisateur telles que Buttons, Images, Texte Boxes, et des fonctionnalités telles que Text-to-Speech, Sensors et GPS.

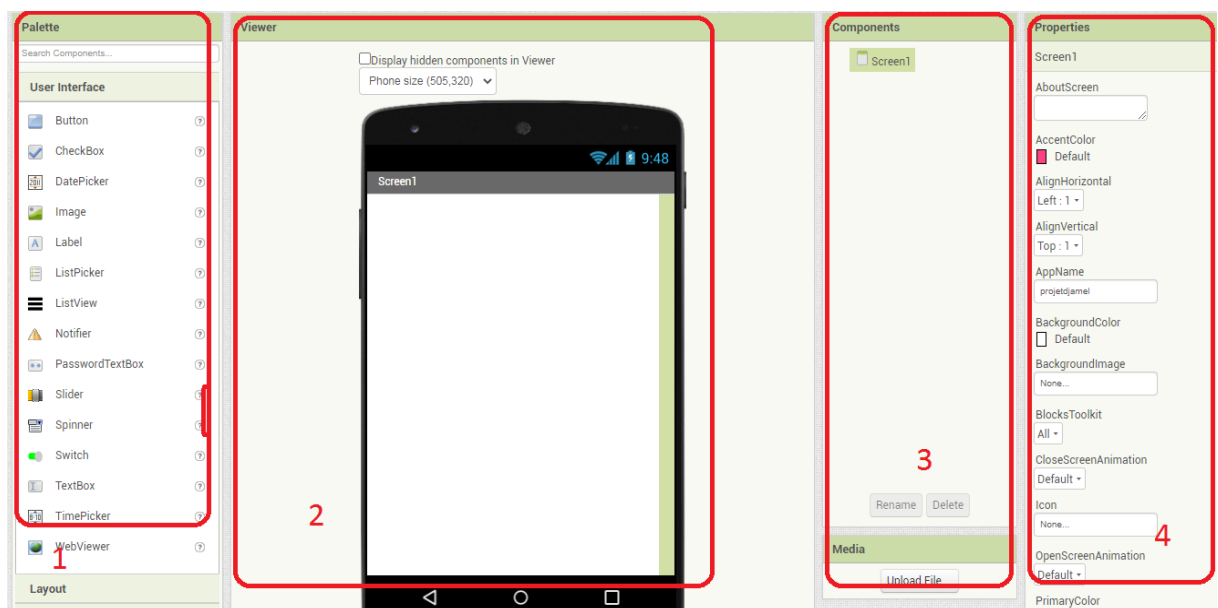


Figure IV.47 : interface de la création App Inventor.

L'interface est composée de quatre fenêtres :

- Fenêtre 1 :c'est la palette, elle contient l'objet disponible.
- Fenêtre 2 :Screen1, c'est la zone du travail.
- Fenêtre 3 :c'est la fenêtre d'exploration, où nous allons trouver la liste des objets déjà choisis.
- Fenêtre 4 :c'est la fenêtre des propriétés, nous pouvons modifier les propriétés des objets sélectionnés.

7. Pour créer le design de notre application, nous utilisons les objets disponibles dans le menu déroulant à gauche de l'écran qui s'appelle Palette et modifier leurs paramètres qui apparaissent à droite de l'écran qui s'appelle propriétés (Figure IV.48) .

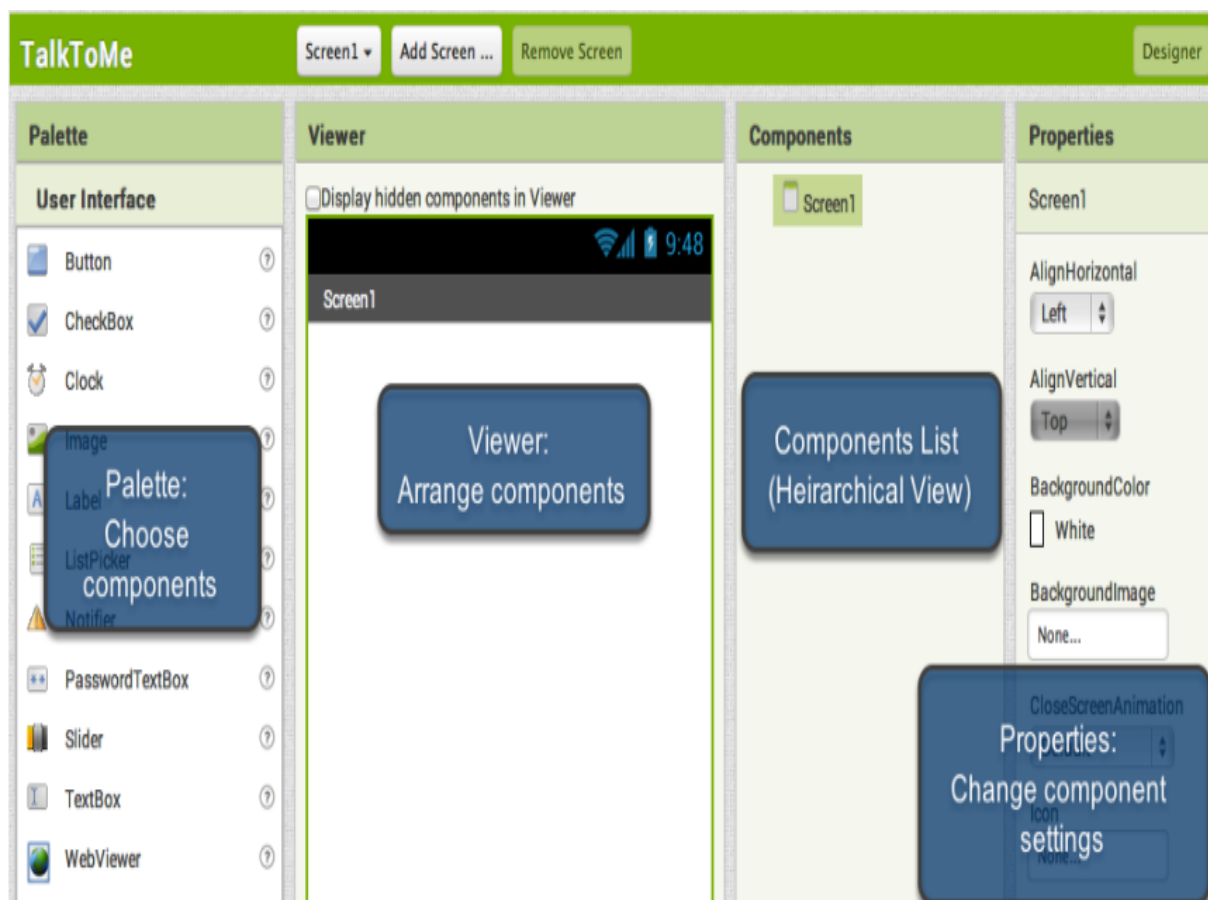


Figure IV.48 : une fenêtre qui nous présente ces composons

8. Pour aligner des objets verticalement, on ajoute un objet **VerticalArrangement** qui l'on trouve dans layout. Dans l'objet VerticalArrangement, placez un label btStatusLabel puis un ListPicker btBtn qui serviront à gérer la connexion Bluetooth (Figure IV.49) .

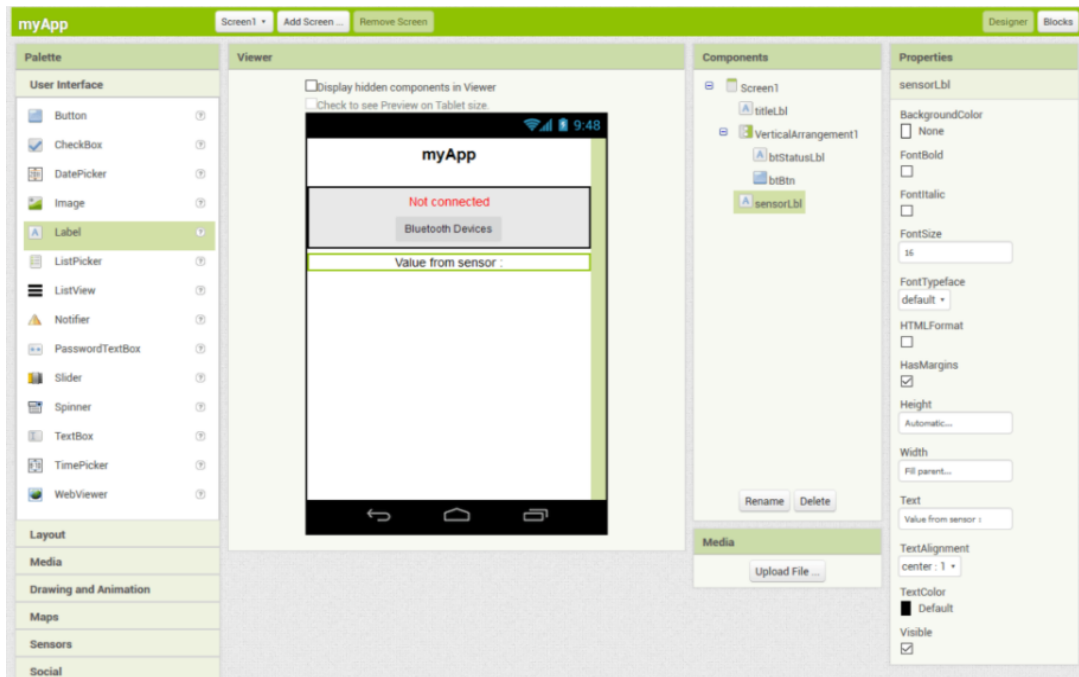


Figure IV.49 : comment aligner des objets

9. Ajouter un **HorizontalArrangement** qui permet de disposer et d'aligner des objets horizontalement et qui contiendra les boutons permettant d'activer la fonction Arduino. Dans l'**HorizontalArrangement**, ajouter deux boutons (Figure IV.50).

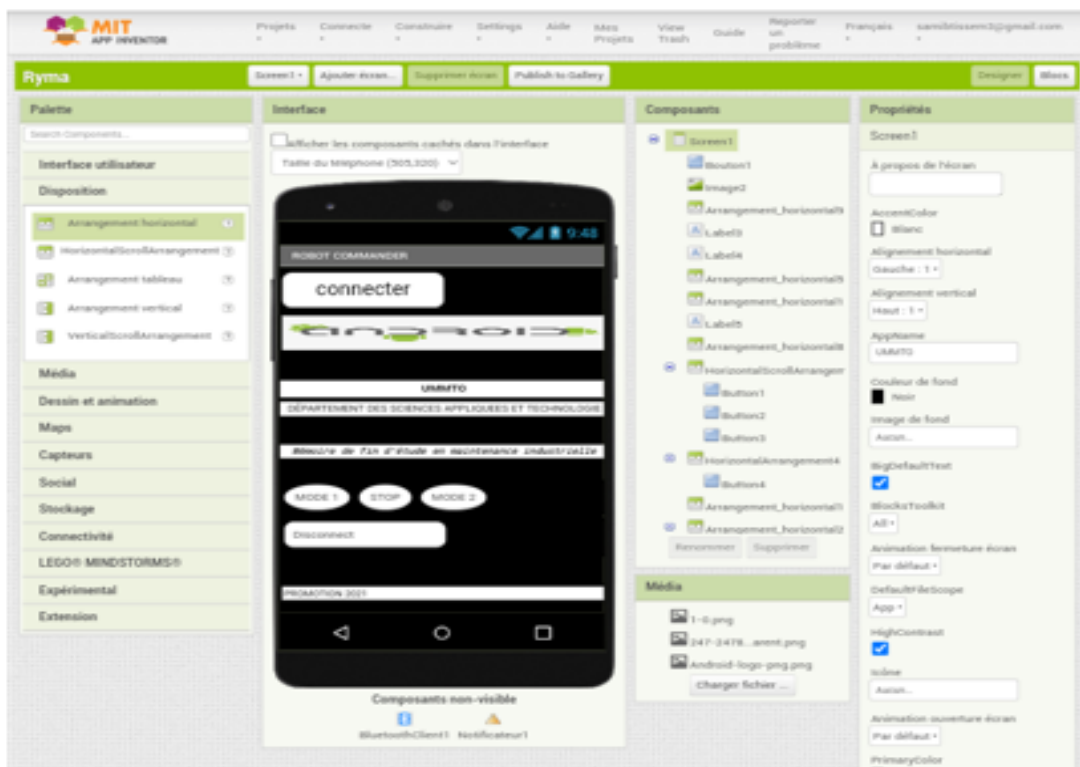


Figure IV.50 : un HorizontalArrangement avec des boutons

10. Pour la Connectivity, ajouter un objet **BluetoothClient** qui nous permettra d'interagir avec le Bluetooth de l'appareil et pour les autres objets on le fait avec les mêmes étapes (Figure IV.51).

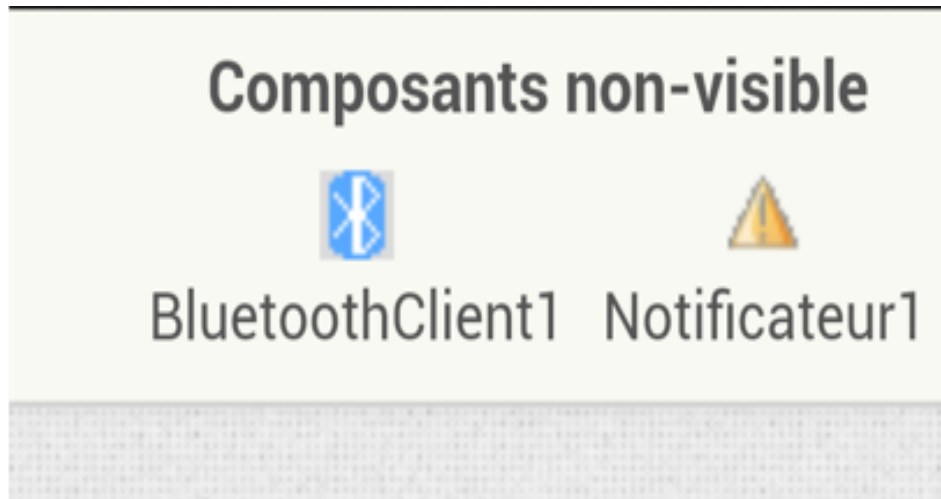


Figure IV.51 : BluetoothClient

11. Enfin nous avons peut personnaliser notre application comme nous le souhaitons en utilisant des couleurs, des polices ou des images différentes (Figure IV.52).

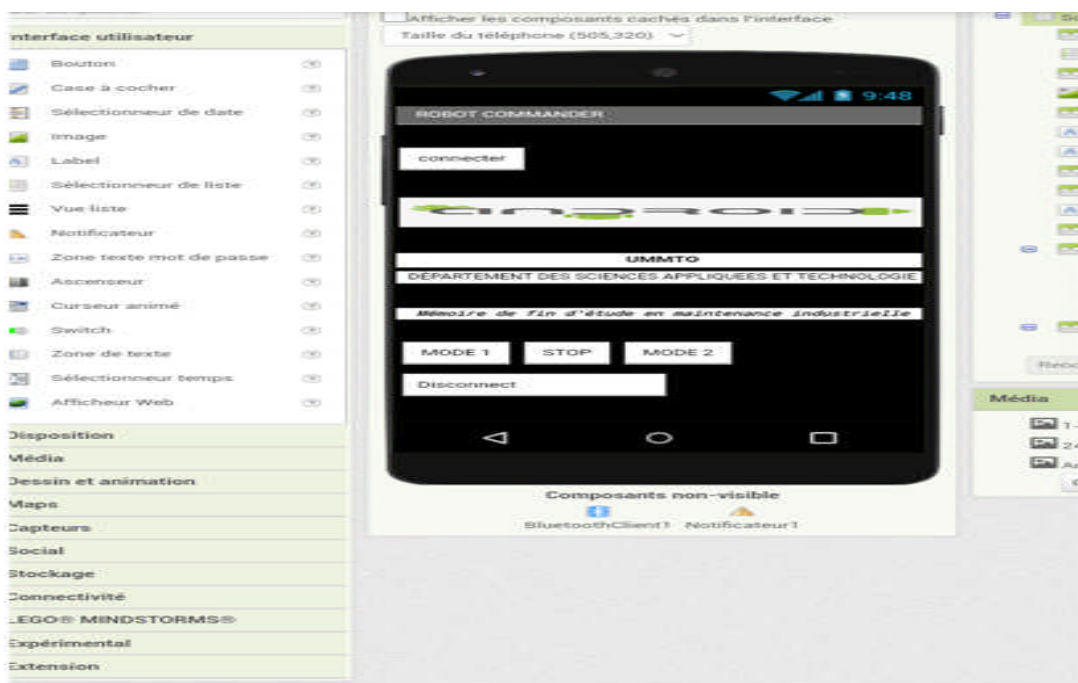


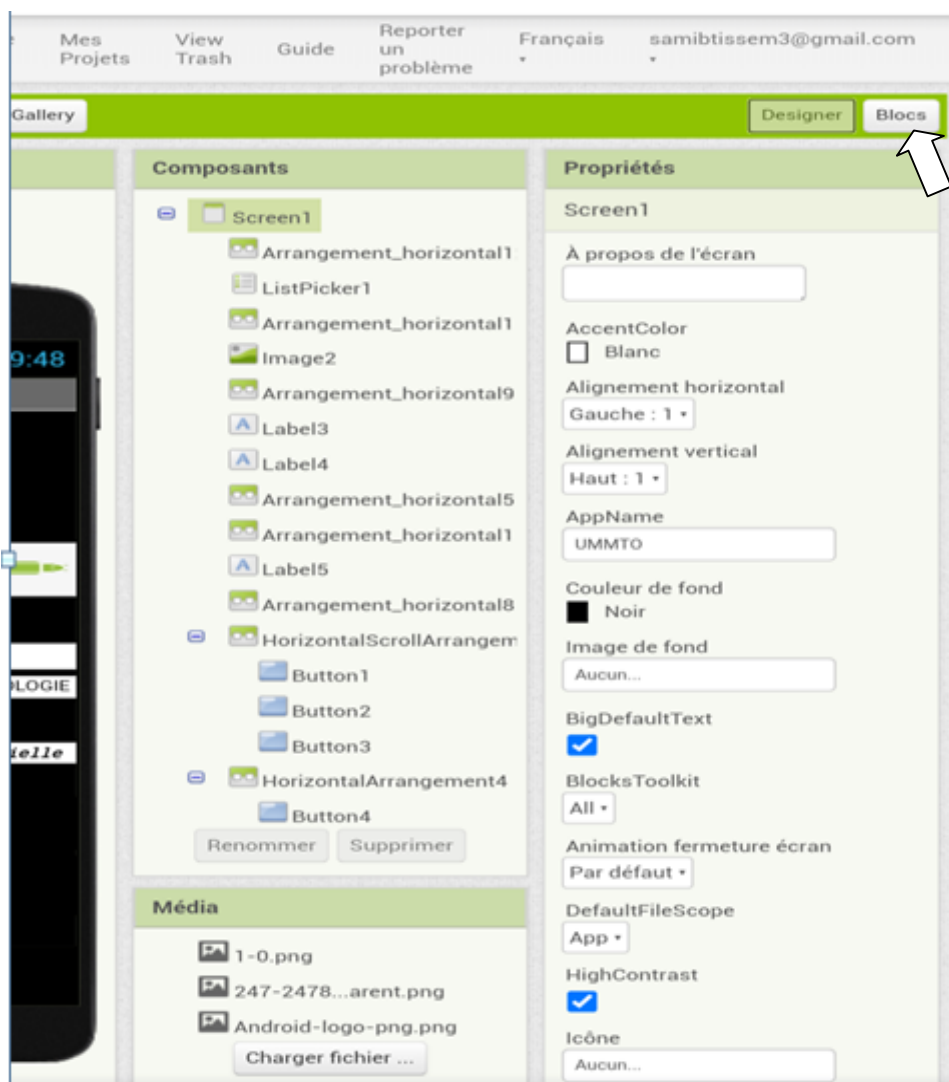
Figure IV.52 :RymaApp

IV.4. Programmation de l'Application AI2

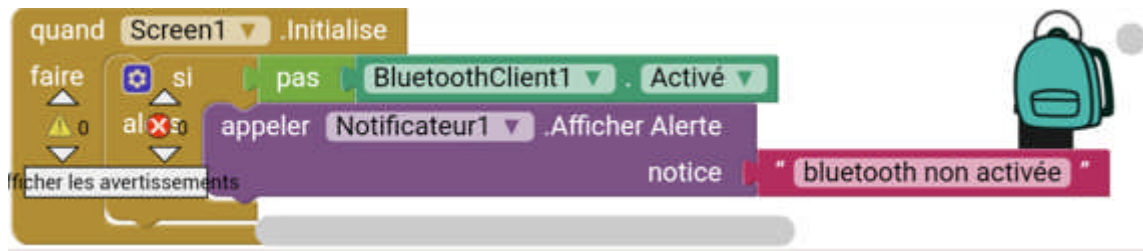
Pour programmer l'application, AI2 nous propose d'utiliser le Blocs qui permet de créer un programme sous forme de schéma bloc. Très simple d'utilisation mais nécessitant un peu de logique de programmation.

Toutes les fonctions sont disponibles en cliquant sur les objets dans le menu déroulant à gauche de l'écran. Parcourez les fonctions des objets que vous avez rajouté sur l'application ainsi que celles des objets Variables, Text, Math et Control dont nous allons nous servir.

1. Cliquons sur "**Blocks**" pour aller sur Blocks Editor. Le Blocks Editor vous permettra de savoir comment l'application doit se comporter. Les boutons Designer et Blocs sont comme des onglets pour passer d'un espace à l'autre sur App Inventor.



2. Initialisation de Screen1 on ajoute **une boucle SI** par ce que le Bluetooth n'est pas active et on va appeler ou Notificateur1.



3. Créez la liste des modules Bluetooth disponibles sur l'appareil.



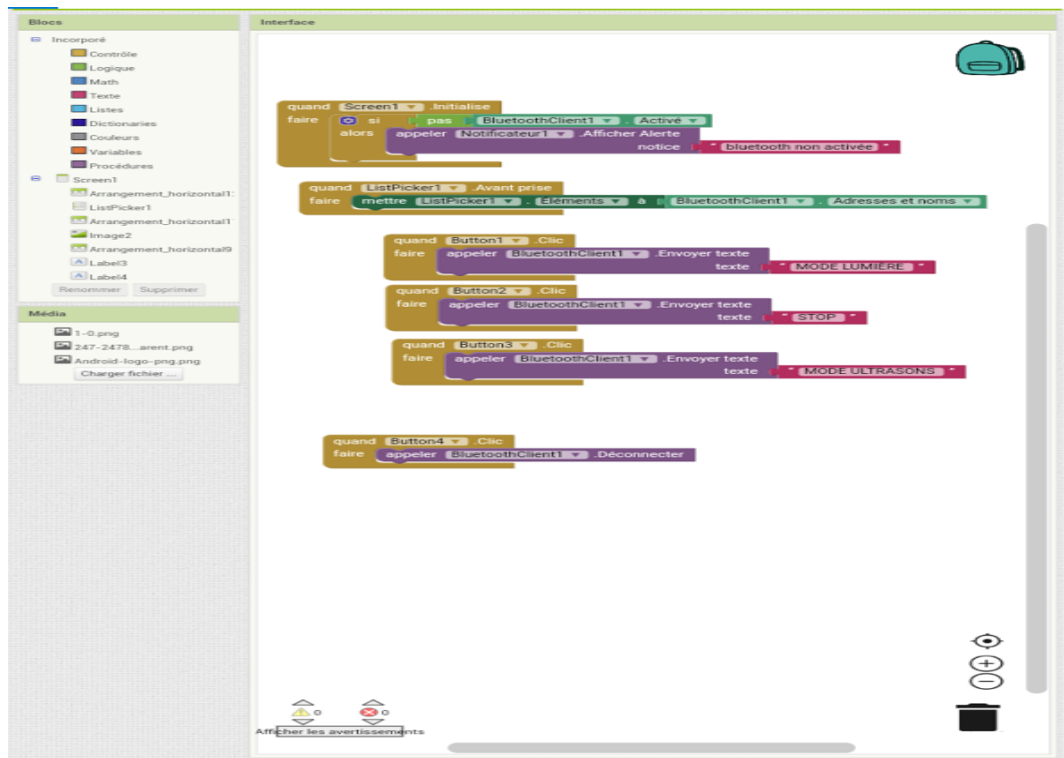
4. Cliquez sur le menu Button1. Cliquez sans relâcher sur le bloc When Button1.Click do. Déplacez-le jusqu'à l'espace de travail et déposez-le. C'est ce bloc qui va gérer ce qui se passe quand on clique sur le bouton de votre appli. Puis Cliquez sur le menu TextToSpeech. . Déplacez le bloc call TextToSpeech1.Speak sur l'espace de travail et déposez-le. Nous appelons ce bloc violet "procedure" dans App Inventor.



5. Enfin , ajout le bloc de bouton de déconnecter .

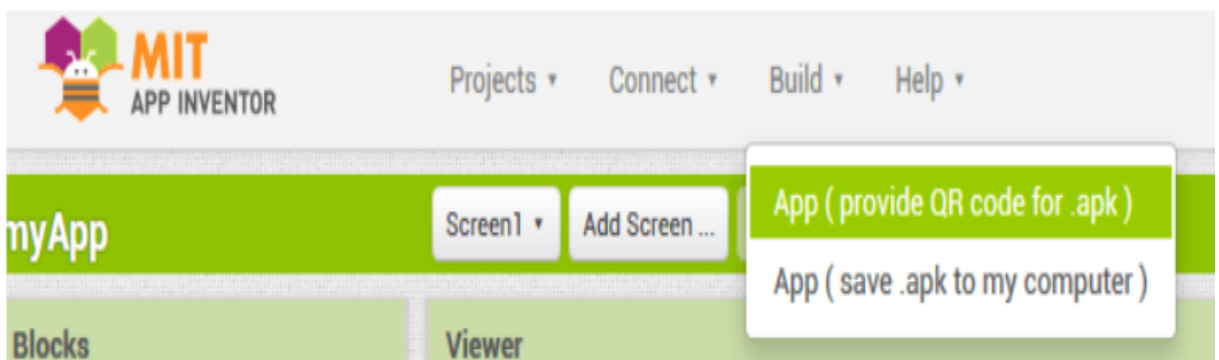


6. Une fois tous les blocs placés sur l'interface et configurés, le code devrait ressembler à ça.

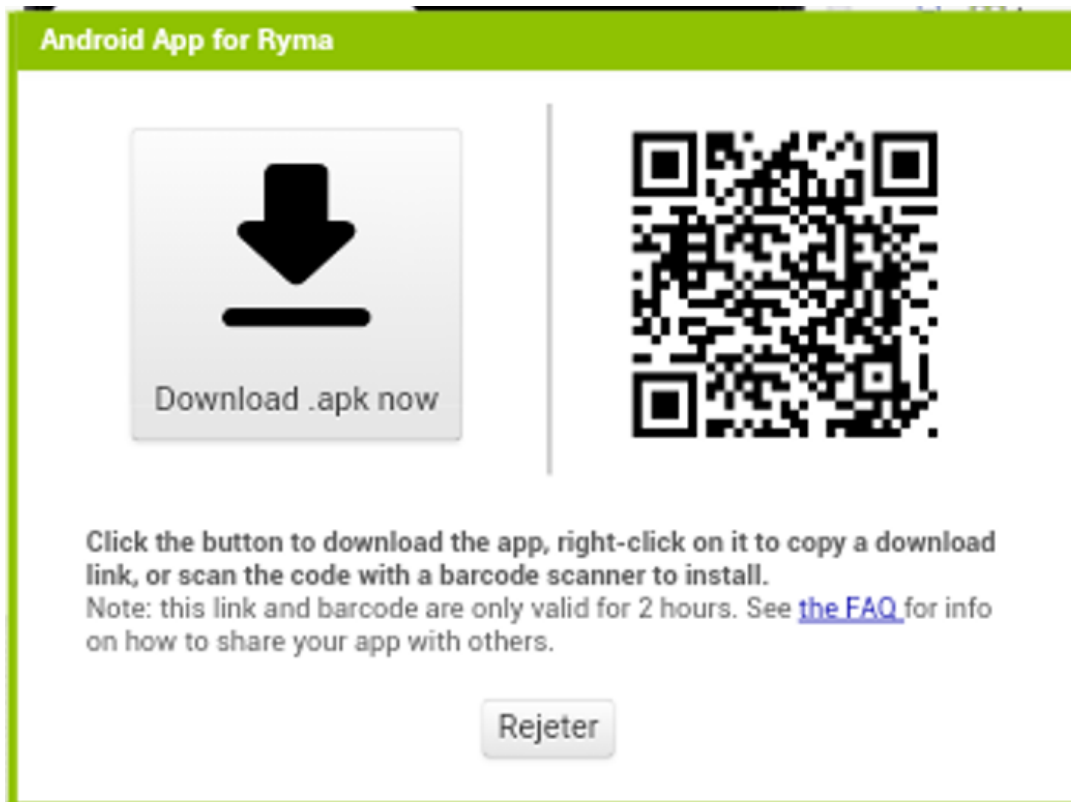


IV.5. Compiler et partager votre Application

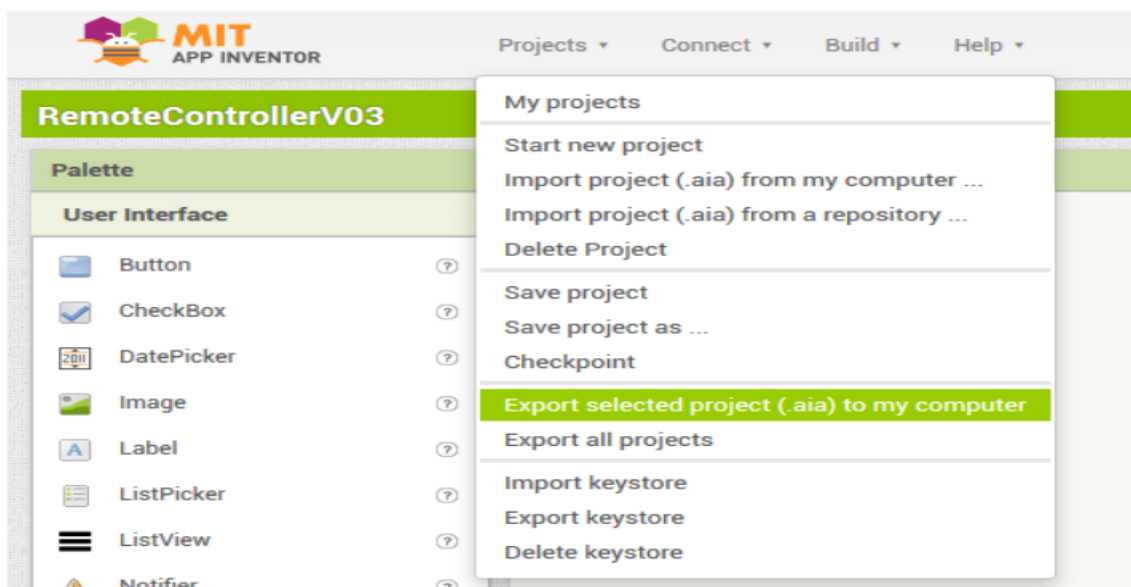
- Une fois que ce programme en Scratch est correctement configuré et qu'aucune erreur ne s'affiche, munissez-vous d'une application de lecture de code QR et compilez le programme en utilisant le menu Build > App (provide QR code for .apk)



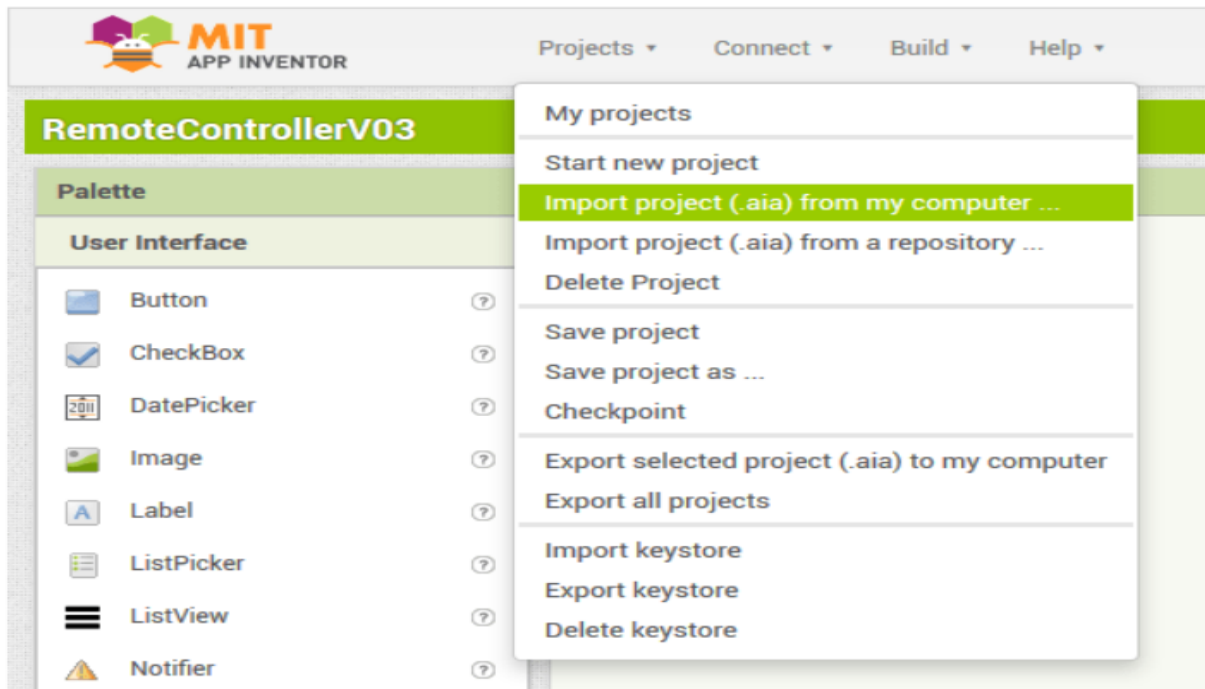
- Une fois le code QR généré, vous pouvez télécharger et installer l'application sur notre Smartphone, on clique sur Download .apk now .



- nous pouvons partager l'installateur de notre application .APK pour Android ou nous pouvons partager le projet .AIA afin que d'autres personnes puissent le modifier.
- Exporter le projet pour créer le fichier AIA correspondant sur notre ordinateur



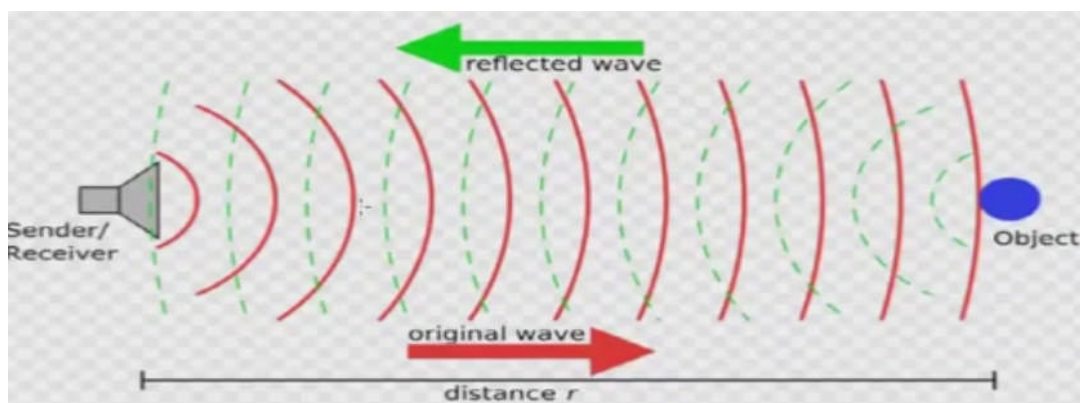
- Importer pour ouvrir un fichier AIA sur l'application en ligne et modifier le projet.



IV. 6. Amélioration de la prissions de robot vé a le capteur ultrason

IV. 6.1. Mesure de la distance par les ULTRASONNS

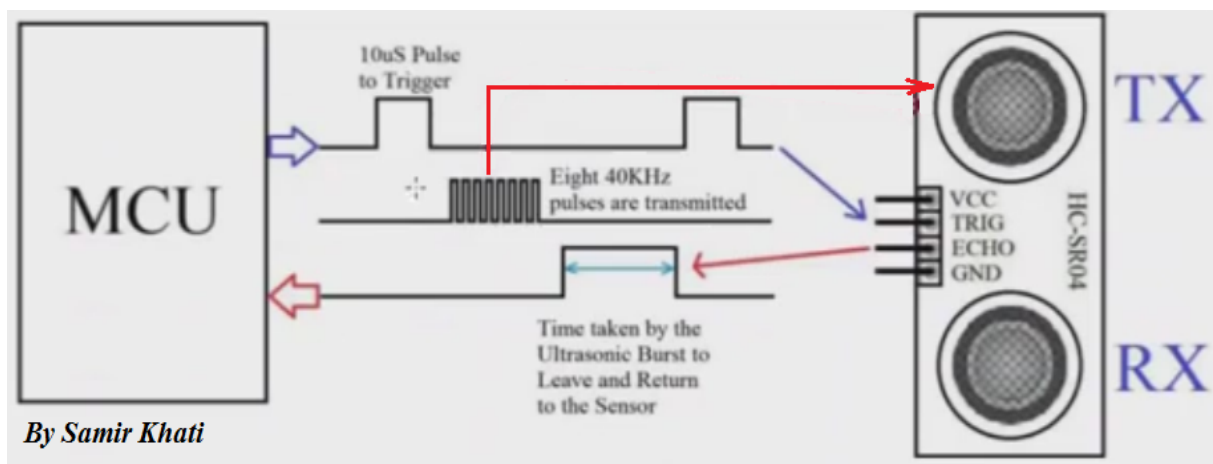
Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur.



L'équation de la propagation du son en fonction de la T° et H

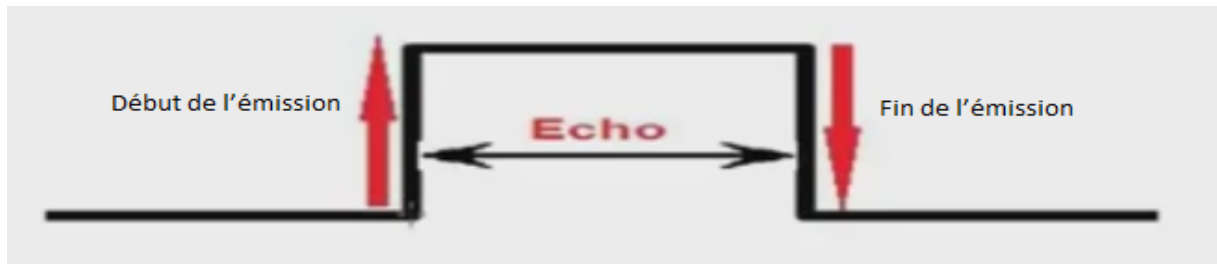
$$C = H + T^\circ$$

- **Trigger pulse input** : Une impulsion de 10 μ s sur cette broche, l'émetteur ultrasonique du module transmet une onde sonore de fréquence 40Khz au même temps le niveau logique sur la broche ECHO passe de niveau logique bas au niveau logique haut. Cette broche pour le module est une entrée (TX), pour le microcontrôleur c'est une sortie.
- **Echo pulse output** : Si l'onde sonore est reçut (réfléchie) par le coaptur (RX), le niveau logique de cette broche retombe a zéro.
- **Vcc** : C'est la broche positive pour alimentation (5V).
- **GND** : Masse.



IV.6.2. Méthode d'application

- Une patte du microcontrôleur doit être configurée en sortie pour fournir une impulsion de 10 μ s a la patte TRIG du module ultra-son.
- Comme c'est déjà dit, lorsque la boche TRIG reçoit une impulsion de 10 μ s, il y'a une émission d'une onde sonore et aussi le niveau logique de la broche ECHO passe de 0 à 1, c'est a partir de ce passage de niveau qu'il faut commencer a compté, c'est-à-dire activé un timer.
- Si l'onde émise est réfléchie, le niveau logique de la broche ECHO revienne à l'état initial (de 1 à 0).
- le temps qu'a prie l'onde sonore pour sa réflexion est déduit par le début de comptage du timer et son arrêt comme le montre la figure :



IV.6.3. Calcul de la distance

D'après le DataSheet du module ultra-son, on donne les formules qui nous permettent de calculer la distance .

En centimètre

$$D = \frac{\text{Timer(uS)}}{58} \text{ Avec Timer (uS) = valeur atteinte (comptage) par le timer lorsque echo = 1}$$

En inch

$$D = \frac{\text{Timer(uS)}}{148}$$

Remarque : L'utilisation de ces deux formules donne un résultat non précis, de ce fait un calibrage (étalonnage) est nécessaire, pour ce la formule citée prend la forme suivante :

$$D = \frac{\text{Timer(uS)}}{58 \times 2}$$

IV.6.4. Application sur Arduino

Connaissant le temps qu'il faut à l'onde ultrasonique pour faire un aller et un retour vers un objet et connaissant aussi la vitesse du son, l'Arduino peut calculer la distance entre ces deux éléments.

La vitesse du son dans l'air change avec le changement de la température et l'humidité. Afin de calculer la distance avec précision, nous devons tenir compte de la température et de l'humidité ambiante.

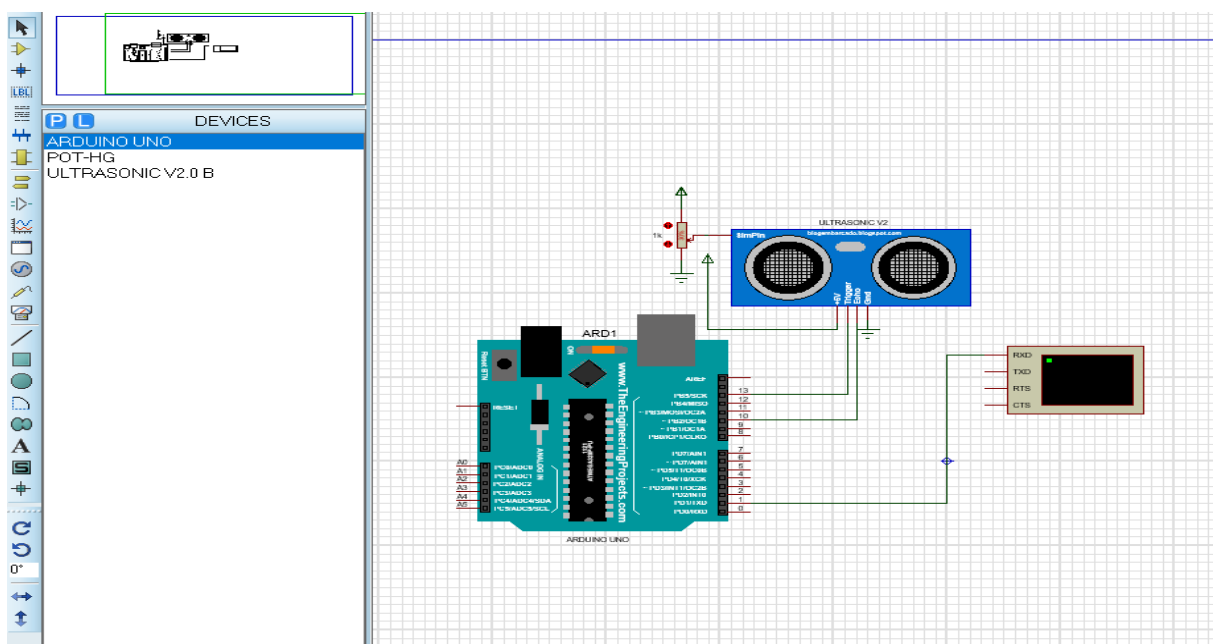
IV.6.5. Programmation

```

1 #define trigPin 13
2 #define echoPin 10
3
4 void setup() {
5   Serial.begin (9600);
6   pinMode(trigPin, OUTPUT);
7   pinMode(echoPin, INPUT);
8 }
9
10 void loop()
11 {
12   float duration, distance;
13   digitalWrite(trigPin, LOW);
14   delayMicroseconds(2);
15
16   digitalWrite(trigPin, HIGH);
17   delayMicroseconds(10);
18   digitalWrite(trigPin, LOW);
19
20   duration = pulseIn(echoPin, HIGH); //c'est pour recevoir la valeur du comptage reçu après
21                                     //la transmission(trigPin).Voir l'explication ci-desou
22
23   distance = (duration / 2) * 0.0344;
24
25   if (distance >= 400 || distance <= 2)
26   {
27     Serial.print("Distance = ");
28     Serial.println("Out of range");
29   }
30   else
31   {
32     Serial.print("Distance = "); //message
33     Serial.print(distance); //valeur calculer
34     Serial.println(" cm"); //message
35     delay(500);
36   }
37   delay(500);
38 }
39

```

IV.6.6. Schématisation sur ISIS PROTEUS



IV.7. Amplification de l'alimentation du Robot

La tension de démarrage automatique sera abaissée à 7 V ou moins, et le moteur passera à haute vitesse lorsque la tension atteindra 15 V ou plus. Il est difficile de travailler avec un équipement électrique 12V, ce module Buck Buck automatique peut résoudre ce problème, quelle que soit la tension d'entrée utilisée (5V, 12V ou 32V), la sortie peut être stabilisée à 12V.

Les commutateurs MOSFET efficaces 4A intégrés permettent une efficacité allant jusqu'à 94%; (LM2577 actuel est 3A)

Haute fréquence de commutation 400KHz, l'ondulation est plus petite, la dimension est plus petite.

IV.7.1. Convertisseur Boost

Dernier convertisseur de la famille des convertisseurs DC/DC le convertisseur buck-boost est également appelé abaisseur-élevateur par son aptitude à fournir une tension plus faible, ou plus élevée que celle appliquée en son entrée. Ceci dépend de la valeur du rapport cyclique. La représentation de ce convertisseur est donnée Convertisseur buck-boost Nous attirons d'emblée l'attention sur une infraction manifeste à une règle établie en première partie de ce cours : un convertisseur direct ne peut mettre en relation deux sources de même nature. La représentation donnée Fig. 1.35 appelle donc à juste titre quelques précisions : – une telle représentation est justifiée dans le sens où le convertisseur DC/DC comporte $\star\star$ en interne $\star\star$ une source de courant telle que, quelle que soit la configuration résultante d'une commutation, les sources de tension d'entrée et de sortie ne sont pas reliées directement. Ceci sera développé ultérieurement. – dans la mesure où l'on définit le buck-boost comme un convertisseur élévateur-abaisseur, le meilleur moyen d'illustrer cela revient à la représentation Fig. 1.35. Nous insistons cependant sur le fait de l'existence, en interne au convertisseur, d'une source de courant permettant de ne pas connecter directement les deux sources de tension U_e et U_s . Par un tel convertisseur, on cherche à fixer une tension moyenne de sortie qui réponde aux critères suivants : – $< U_e$ ou bien $> U_e$, – réglable à souhait dans l'une des deux fourchettes définies ci-dessus, sachant que l'on doit pouvoir passer sans discontinuité d'une fourchette à l'autre. La source d'entrée du **convertisseur DC DC boost** est une tension continue. La bobine fait le plus gros du travail, car elle va accumuler de l'énergie sous forme d'énergie magnétique. Vous avez également l'interrupteur qui est

généralement un transistor à découpage. Ce transistor va rapidement commuter pour passer de la phase accumulation à la phase restitution.

De plus, vous avez une diode et un condensateur de filtrage pour filtrer le signal. Une tension continue n'est jamais parfaite, car il existe quelques petites oscillations. Le but de ce condensateur est de réduire un maximum ces oscillations, afin de fournir un signal le plus propre possible à la charge. Pour en savoir plus sur ce convertisseur DC DC, veuillez lire l'article adéquat.

IV.7.2. Pacifications

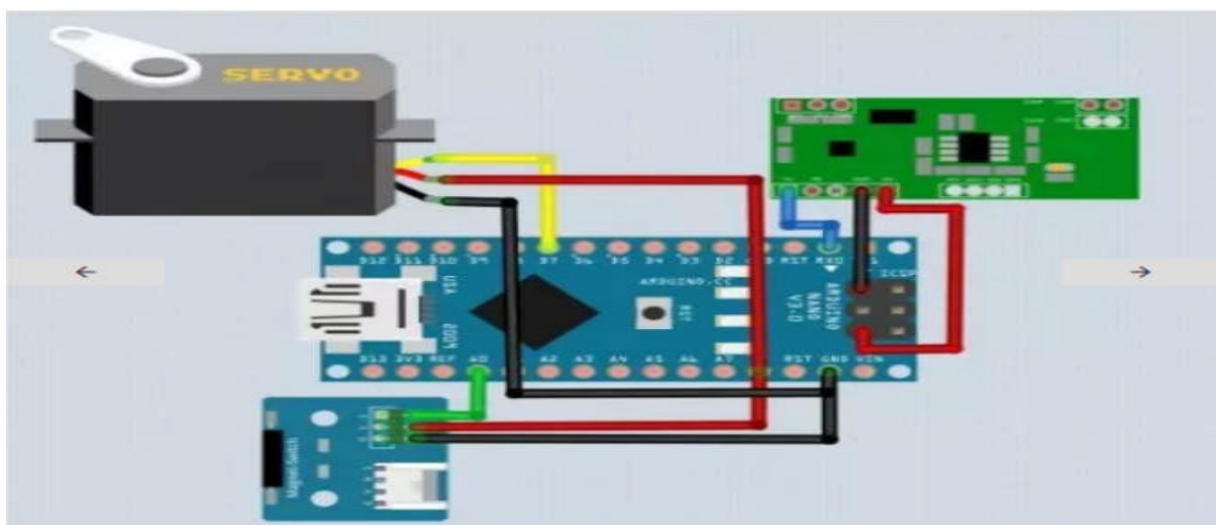
- Plage de tension d'entrée: 3,5 V-18V (la tension de sortie doit être défini comme supérieure à la tension d'entrée)
- Plage de tension de sortie: 4,0 V-24V
- Courant d'entrée maximal: 3A (le courant de sortie dans des conditions nominales est calculé sur la base de la conservation de l'énergie, $U_{in} * I_{in} * \text{Efficacité} = U_{out} \text{ out}$)
- Entrée continue du courant de travail: 2A (la tension de sortie pour le rapport de la tension d'entrée doit être inférieur à 3 dans de bonnes conditions d'aération)
- Dimensions: L * W * H = 4.3cm * 3.0cm * 1.2cm

IV.7.3. Exemple de code

```
sketch_jan16a $
1 //Chaîne comdata = "";
2 int sortie=2;
3
4 void setup ()
5 {
6 pinMode (2,sortie);
7 pinMode (3,sortie);
8 digitalWrite (2, HIGH);
9 digitalWrite (3, HIGH);
10 Serial.begin (9600);
11 }
12
13 void loop ()
14 {
15
16 digitalWrite (2, HIGH);
17 digitalWrite (3, HIGH);
```

```
16 digitalWrite (2, HIGH);
17 digitalWrite (3, HIGH);
18 while(Serial.disponible ()> 0);
19 {
20 Comdata += char(Serial. le lire ());
21 delay (2);
22 }
23 Si (comdata. longueur ()> 0 & & comdata == "0500 addf1443").
24 {
25 While (comdata. longueur ()> 0);
26 {
27 digitalWrite(3, LOW);
28 delay(100);
29 While (Serial.disponible ()> 0);
30 {
31 Comdata += char (Serial. le lire ());
32 delay (2);
33 }
34 Comdata = "";
35 }
36 }
37 else if (comdata.longueur ()> 0);
38 {
39 While (comdata.longueur ()> 0);
40 {
41 digitalWrite (2, LOW);
42 delay (100);
43 While (Serial.disponible ()> 0);
44 {
45 Comdata += char (Serial.lire ());
46 delay (2);
47 }
48 Comdata = "";
49 }
50 }
51 }
52 }
```

IV.3.4. Exemple de câblage



Conclusions

À la fin de ce chapitre, nous devrions être capable de :

- Explorer les onglets "design" et "blocks" d'App Inventor afin de construire ce dont vous avez besoin.
- Tester vos fonctions : c'est génial de pouvoir bâtir une fonction mais encore faut-il qu'elle marche correctement.
- Toujours penser comment vous pourriez vérifier une deuxième fois que ce que vous faites est bien ce que vous pensez faire.
- Progresser pas à pas : les fonctions sont souvent construites l'une après l'autre. Ainsi, si un élément de votre projet casse, vous trouverez plus facilement où peut-être le problème.
- Nous avons amélioré la puissance de robot grâce au capteur ultrason et amplifié de l'alimentation du Robot avec la boost.

Chapitre 4
AMELIORATI
-ON
DU ROBOT

Conclusion générale

Conclusion générale

La mise en marche d'un robot mobile nécessite une étude sur plusieurs domaines différents : l'électronique, la mécanique et l'informatique .ce qui m'a donné l'opportunité de travailler sur plusieurs domaines à la fois.

Dans notre travail, nous avons conçu et réalisé un robot mobile à base d'une carte Arduino télécommandé à distance via une application Android « MIT APP INVENTOR », avec laquelle nous allons choisir le mode de contrôle : autonome ou en fonction de l'intensité de la lumière ou avec une caméra.

Une transmission Bluetooth qui assure la communication entre la carte Arduino et l'application Android.

Une camera pour visualiser et surveiller à distance et en temps réel l'environnement qui transmet l'image vers son écran.

Une création d'une application personnel.

La première partie de ce mémoire est consacrée à donner une vision générale sur la robotique et les robots mobiles.

Dans la seconde partie, nous avons présenté le matériel utilisé dans notre système, commençant par la carte Arduino et les capteurs connectés à cette carte, et aussi donné une description des moteurs et les logiciels utilisés , et la troisième partie nous avons introduit une introduction général sur l' asservissement Visual d'un robot.

A la fin de cette partie, on a présenté l'application Android et les différentes étapes pour sa création +une amélioration d'un robot avec un boosteur et un ultrason.

La dernière partie de ce mémoire est consacré à la conception et la réalisation de l'application où nous avons illustré les différentes étapes avec des images.

De manière générale, ce travail m'a fait découvrir le monde de la robotique en général, spécialement la robotique mobile et ses perspectives.

A la fin de ce projet m'a permis de me familiariser avec la programmation physique, il m'a donné des idées sur la création des applications Android et cela en appliquant les connaissances acquises durant mon cursus universitaire.

En réalisant ce projet, j'ai pris conscience de l'existence des imperfections, j'ai donc pensé à des perspectives d'améliorations et d'autres idées permettant de le compléter.par exemples :

- ✓ ajouter des capteurs tels que le capteur de température et humidité, capteur à gaz et fumée...
- ✓ utiliser une camera programmable afin d'avoir l'image sur l'interface de l'application de la télécommande.
- ✓ Commander le robot avec une reconnaissance vocale.

Bibliographie

Bibliographie

- [1] : Dictionnaire encyclopédique en un volume, publié en 1905 sous la direction de Claude Augé (Petit Larousse illustré).
- [2] : Nicole Merle-Lamoot, Gilles Pannetier, Le développement industriel futur de la robotique personnelle et de service en France,2012.
- [3] :file:///C:/Users/SyscoMPUTER/des Ktop/mimmoire/chapitre1.pdf.
- [4] : Laetitia Matingnon<<Introduction a la robotique>> Université de Caen Bass-Normandie,Licence 1 ere année-2011/2012.
- [5] : Omar ICHEROUFENE, «Conception et réalisation d'un Robot mobile télécommandé à base de la PCDUINO V3», Mémoire de fin d'étude, Université UMMTO, 2016.
- [6] : <https://fr.wikipedia.org/wiki/domotique>. Consulté le 25-12-2020.
- [7] : MOUSSAOUI Amira, « Conception et réalisation d'un bras manipulateur commandé par l'Arduino Méga 2560 », Mémoire de fin d'étude, Université M'hamed Bougara, 2017.
- [8] : S Landrault , H Weisslinger ;Arduino : Premiers pas en informatique embarquée ; Édition du 19 juin 2014.
- [9] : <https://fr.wikipedia.org/wiki/domotique>. Consulté le 25-12-2020
- [10] : S Landrault , H Weisslinger ;Arduino : Premiers pas en informatique embarquée ; Édition du 19 juin 2014.
- [11] : <https://fr.wikipedia.org/wiki/domotique>. Consulté le 25-12-2020.
- [12] : S Landrault , H Weisslinger ;Arduino : Premiers pas en informatique embarquée ; Édition du 19 juin 2014.
- [13] : <https://fr.wikipedia.org/wiki/domotique>. Consulté le 25-12-2020.
- [14] :w.w.w.robot-maker.com/shop/capteurs/31-dht22.html.
- [15] : Anonyme ; « Fiche F3 – Contrôler Des Moteurs Et Diriger Un Robot.pdf ».
- [16] : M.Lakhdari .F, « Introduction à la simulation et routage des circuits avec le logiciel PROTEUS V7 et V8 ». Université d'ORAN ,2017.
- [17] :YuTo-ESP32-CAM-Premiers Pas/ L.Chastain/AC.Limoges-Janv.2020.
- [18] : <https://w.w.w.hwlibre.com> .Cnd.ampproject.org/V/S/ www.hwlibre.com/fr.
- [19] : M.Lakhdari .F, « Introduction à la simulation et routage des circuits avec le logiciel PROTEUS V7 et V8 ». Université d'ORAN ,2017.
- [20] : [Rémi10]: R. Bouteau. Thèse de doctorat "Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions" _ l'Université de ROUEN "; 19 avril 2010.
- [21] : [Barbosa04]: G. Barbosa, J.J, Thèse de doctorat " Vision panoramique pour la

Bibliographie

robotique mobile : stéréovision et localisation par indexation d'images" , Université Toulouse III, 2004.

[22] : [Yagi99]: Y. Yagi. Omnidirectional Sensing and Its Applications, IEICE Transactions on Information and Systems, vol. E82-D, pp. 568-579, 1999

[23] : [David et al]: F. David, E. Mouaddib J. Salvi, " Segmentation et Décodage d'un Patron de Lumière Structurée ".CREA, Université de Picardie Jules Verne, Amiens, France, david.fofi@u-picardie.fr II, Université de Girona, Espagne, qsalvi@eia.udg.es

[24] : [Aurore13] : A. Aurore, thèse de doctorat "Squelettisation en un balayage Application à la caractérisation osseuse ", version 1. 25 janvier 2013. tel-00781222

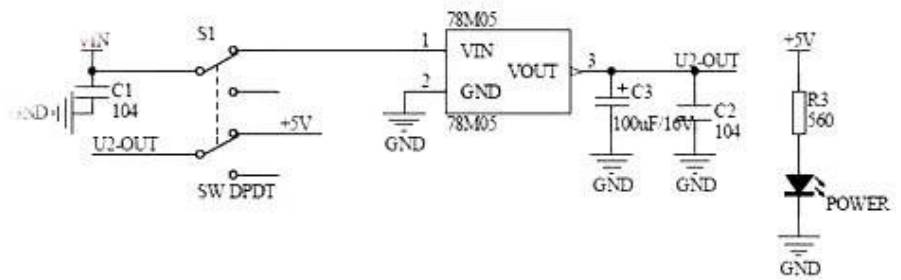
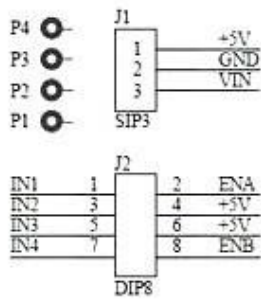
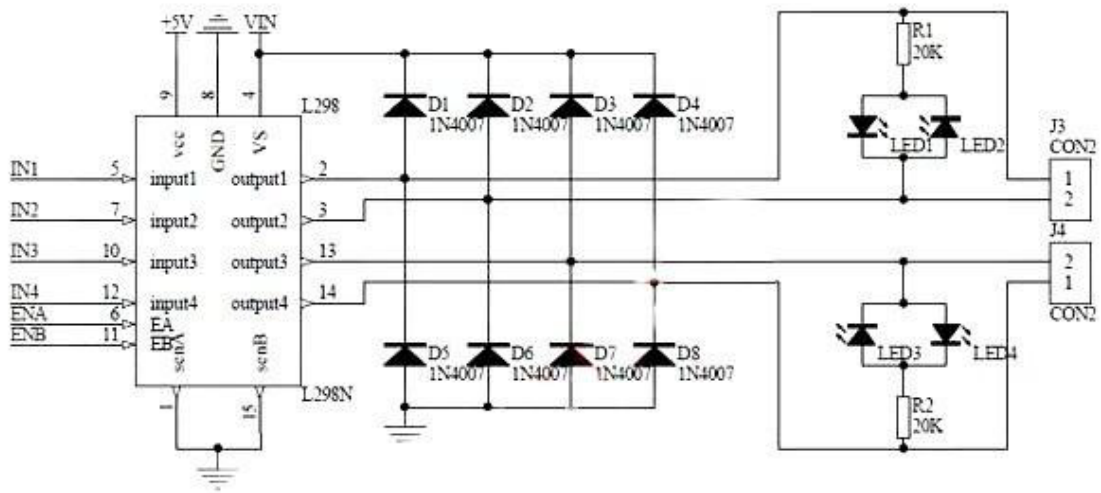
[25]: [Lam et al92] : L. Lam, Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, September 1992, page 879, bottom of first column through top of second column

[26] : Adrien Durand-Petiteville. Navigation référencée multi-capteurs d'un robot mobile en environnement encombré. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2012. Français. fftel006943

[27] : [Morph]: http://sparis.free.fr/Cours_Image/AI6_1Operateurs_Morphologiques . ppt

Annexe

Schéma du pont H L298N



Carte Arduino

